

AD-A147 716 FINPLAN: A PROTOTYPE EXPERT SYSTEM FOR PERSONAL
FINANCIAL PLANNING(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH J D PINC AUG 84

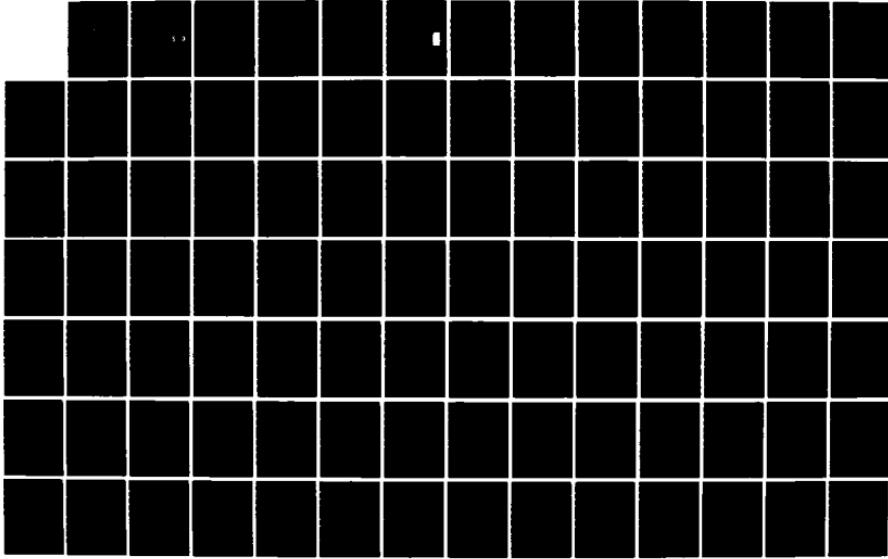
1/2

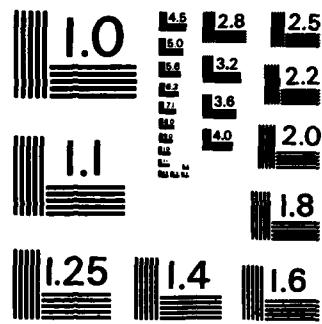
UNCLASSIFIED

AFIT/C/NR-84-67T

F/G 9/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

AD-A147 716

DTIC FILE COPY

REPORT DOCUMENTATION PAGE			READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 84-67T	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) FINPLAN: A Prototype Expert System For Personal Financial Planning		5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION	
7. AUTHOR(s) James D. Pinc		6. PERFORMING ORG. REPORT NUMBER	
8. CONTRACT OR GRANT NUMBER(s)		9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: University of Florida	
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS		11. CONTROLLING OFFICE NAME AND ADDRESS IT/NR AFB OH 45433	
12. REPORT DATE 1984		13. NUMBER OF PAGES 98	
14. SECURITY CLASS. (of this report) UNCLASS		15. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) PROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) S DTIC ELECTE NOV 19 1984 D B			
18. SUPPLEMENTARY NOTES APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1 LYNN E. WOLAYER Dean for Research and Professional Development AFIT, Wright-Patterson AFB OH 12 Oct 84			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)			
ATTACHED			

84 11 14 147

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

**FINPLAN: A PROTOTYPE EXPERT SYSTEM
FOR PERSONAL FINANCIAL PLANNING**

By

JAMES D. PINC

August 1984

Chairman: Douglas D. Dankel, II, Ph.D.
Major Department: Computer and Information Sciences

FINPLAN is a preliminary prototype program performing personal financial planning services. It is implemented primarily as a production system using expert system concepts and runs on an IBM-compatible personal computer with 128K bytes of memory. Two principal tasks of financial planning are addressed: determining a sufficient amount of life insurance coverage and providing recommendations regarding an investment portfolio for a given client-user. The determination of life insurance needs represents a class of numerically oriented problems, which are common in the domain of personal financial planning. FINPLAN offers a solution to this class of problems by integrating numerical algorithms with

production rules. Recommendations regarding the client's investment portfolio are based primarily on inference procedures and other expert system concepts. FINPLAN selects appropriate investment classes by matching the characteristics of each investment class with the client's own temperament and goals. The thesis also gives a general description of rule-based expert systems, discusses selected design features considered in the development of FINPLAN, and suggests potential improvements that may lead to a future, more comprehensive and usable financial planning system.



Chairman

Accession For	
NTIS	GRA&I <input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (AFIT). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

RESEARCH TITLE: FINPLAN: A Prototype Expert System For Personal Financial Planning

AUTHOR: James D. Pinc

RESEARCH ASSESSMENT QUESTIONS:

1. Did this research contribute to a current Air Force project?

a. YES b. NO

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

a. YES b. NO

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

a. MAN-YEARS _____ b. \$ _____

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3. above), what is your estimate of its significance?

a. HIGHLY SIGNIFICANT b. SIGNIFICANT c. SLIGHTLY SIGNIFICANT d. OF NO SIGNIFICANCE

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the bottom part of this questionnaire for your statement(s).

NAME _____ GRADE _____ POSITION _____

ORGANIZATION _____ LOCATION _____

STATEMENT(s):

FOLD DOWN ON OUTSIDE - SEAL WITH TAPE

AFIT/NR
WRIGHT-PATTERSON AFB OH 45433

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, \$300



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 72226 WASHINGTON D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

AFIT/ DAA
Wright-Patterson AFB OH 45433



FOLD IN

67

**FINPLAN: A PROTOTYPE EXPERT SYSTEM
FOR PERSONAL FINANCIAL PLANNING**

BY

JAMES D. PINC

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

1984

TABLE OF CONTENTS

	PAGE
ABSTRACT.	iv
CHAPTER	
I INTRODUCTION.	1
Overview.	1
Expert System Defined	2
Personal Financial Planning Defined	3
Reasons for a Financial Planning Expert System	4
FINPLAN	5
II BACKGROUND INFORMATION--A GENERAL DESCRIPTION OF EXPERT SYSTEMS.	7
Overview.	7
Components of Expert Systems.	7
Conclusion.	16
III OVERVIEW OF FINPLAN	17
General Comments.	17
Estimating Life Insurance Needs	20
Making Investment Portfolio Recommendations .	27
IV SPECIAL DESIGN CONSIDERATIONS AND ALTERNATIVES	34
Introduction.	34
Design Considerations and Alternatives.	34
V SUGGESTED IMPROVEMENTS FOR FINPLAN.	45
Introduction.	45
Limitations and Improvements.	45
LIST OF REFERENCES.	51
APPENDICES	
A LIST OF FILE CONTENTS	53

B	FUNCTION DESCRIPTIONS54
C	SOURCE CODE LISTINGS.59
	BIOGRAPHICAL SKETCH99

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

FINPLAN: A PROTOTYPE EXPERT SYSTEM
FOR PERSONAL FINANCIAL PLANNING

By

JAMES D. PINC

August 1984

Chairman: Douglas D. Dankel, II, Ph.D.
Major Department: Computer and Information Sciences

FINPLAN is a preliminary prototype program performing personal financial planning services. It is implemented primarily as a production system using expert system concepts and runs on an IBM-compatible personal computer with 128K bytes of memory. Two principal tasks of financial planning are addressed: determining a sufficient amount of life insurance coverage and providing recommendations regarding an investment portfolio for a given client-user. The determination of life insurance needs represents a class of numerically oriented problems, which are common in the domain of personal financial planning. FINPLAN offers a solution to this class of problems by integrating numerical algorithms with

production rules. Recommendations regarding the client's investment portfolio are based primarily on inference procedures and other expert system concepts. FINPLAN selects appropriate investment classes by matching the characteristics of each investment class with the client's own temperament and goals. The thesis also gives a general description of rule-based expert systems, discusses selected design features considered in the development of FINPLAN, and suggests potential improvements that may lead to a future, more comprehensive and usable financial planning system.



Chairman

CHAPTER I INTRODUCTION

Overview

Research in the computer science discipline of artificial intelligence (AI) has received increasing interest since the mid-1960's. Some of AI's recent success can be attributed to the development of expert systems, which are high-performance programs that solve problems requiring significant human expertise. An expert system differs from a conventional data processing system primarily due to its reliance on knowledge-based inferences as the problem-solving technique.

The first useful expert systems did not emerge until the mid-1970's. Today scores of systems are under serious development or are being used on an experimental basis. A predominance of existing systems perform tasks in the domains of medicine and chemistry. Other systems address problems ranging from configuring computers to constructing other expert systems, themselves [2]. No research references, however, cite any efforts to develop an expert system in the domain of personal financial planning. Furthermore, no major existing system is designed to run on a personal computer.

Because of these gaps in the research on expert systems, this thesis effort sought to develop a prototype expert system performing personal financial planning on a personal computer. Remaining chapters of the thesis revolve around this prototype; the design, methodologies, and limitations of the prototype system are discussed.

Expert System Defined

A consensus of practitioners in the AI field probably would agree an expert system is an intelligent computer program solving problems in a narrowly defined domain of expertise using a body of knowledge and inference procedures. This definition implies several key characteristics of expert systems. First, the problems to be solved must be difficult enough to require significant human expertise for its solution. Second, the body of knowledge, often expressed as rules of reasoning, is originally acquired and developed by a human expert. Last, the knowledge necessary to perform at a high level, plus the inference procedures used, can be considered as a model of the expertise of the best practitioners in the field. Hence, expert systems aim not only to simulate human thought processes, but also to perform as well or better within its designated domain than its human counterpart.

Another key feature of expert systems is their ability to explain output. The capability for explanation separates expert systems from conventional computer

programs using numerical algorithms. Statistical regression analysis programs, for example, perform well on many important problems, but rehashing complicated mathematical formulae offers little illumination of the reasoning behind the solution. The inferences used by expert systems do not necessarily embody exact psychological models of the reasoning of experts, but they do provide understandability to persons familiar with the problem [1].

Chapter II describes the general nature of expert systems in more detail.

Personal Financial Planning Defined

Personal financial planning is the process of designing an overall program to guide the financial dealings of an individual or family. Financial goals are first defined and then pursued through an integrated program of savings, insurance, and investments.

The most central goal in any financial plan should be the provision of financial security. It is one of the most important life-time objectives for an individual or family in Western societies. Achieving financial security means not only the accumulation of future wealth through an investment portfolio, but also the establishment of an immediate estate in the event of adversity. Such adversity might include untimely death, interruption of income, or any other unforeseen happening that may place a claim on an individual's assets or the ability to earn and use income.

A variety of social insurance programs within the United States provides some means of ensuring personal financial security. In most cases, however, benefits derived from these programs are grossly inadequate or fail to cover all persons. A general strategy for achieving financial security usually includes two broad initiatives. First, a carefully designed, self-initiated investment portfolio must be undertaken in order to augment social or other retirement benefits and provide financial security in old age. The immediate need for financial security, on the other hand, must be provided by privately-owned insurance programs integrated with other social and employment-related benefits. Such insurance must include protection against financial loss in the areas of liability, life, health, and property.

In addition to determining the best-suited investments for a portfolio and the types and coverages of private insurance, financial planning also encompasses the provision of tactical advice. Expert advice helps to facilitate an individual's pursuit of financial security. Tax planning, cash budgeting, credit management, and banking services represent some areas of special advisement related to financial planning.

Reasons for a Financial Planning Expert System

Expert system concepts can provide a superior means for solving personal financial planning problems. Several reasons explain why financial planning represents an appropriate domain for the expert system paradigm.

First, financial planning is an area of human expertise where heuristics and specialized knowledge can be represented effectively by production rules. Solutions to financial planning problems do not involve exclusive execution of numerical algorithms, but largely include inferences of deductive reasoning. A complete set of rules of knowledge includes many subtle inferences that only can be imparted to a system by an expert within the field. Successful incorporation of such rules would make a financial planning expert system an extremely useful and valuable computer program.

The facility for explanation, an exclusive feature of expert systems, provides a second reason for performing financial planning using expert system concepts. Financial advice often can not be offered the credibility and support of rigorous mathematical formulae, but it can be supported by a description of the lines of reasoning used to deduce it. Expert systems excell at providing an explanation of their deductions. Inferences that can be displayed for critical review by the user lend both credibility and understandability to the system's output.

Finally, personal financial planning represents an inherently important and critical human endeavor within Western societies. Yet, the majority of United States citizens remain largely uninformed about personal finances. A financial planning expert system can provide both a means of learning about financial planning as well

as an inexpensive and effective source of expert knowledge for designing and executing personal financial plans.

FINPLAN

A major part of the thesis effort involved the development of a prototype expert system that performs personal financial planning. FINPLAN is the name given to this expert system program. It is not a complete program; but it does demonstrate key methodologies that might eventually be used in a more comprehensive and complete expert system. Chapter III describes the architectures developed for FINPLAN. Chapters IV and V offer discussions of selected design considerations and ideas for future system improvements, respectively.

CHAPTER II BACKGROUND INFORMATION--A GENERAL DESCRIPTION OF EXPERT SYSTEMS

Overview

Expert systems represent a relatively immature branch of AI where a consensus of clearly defined concepts and characterizations does not exist. This chapter consolidates a variety of views, and briefly summarizes what expert systems are and how they work. Most expert systems, in some form or another, consist of a knowledge base, global data base, inference engine, justifier, and language processor. The knowledge base represents the source of all rules of reasoning; the inference engine drives and controls interpretation of the rules; the justifier explains the system's rationale in making decisions; and the language processor guides interaction between the user and the system. Each of these components will be briefly discussed.

Components of Expert Systems

The knowledge base

The body of knowledge used by an expert system incorporates its expertise in the form of facts and heuristics. The facts constitute information which is generally agreed upon by experts in the field, and are publicly available. The heuristics, however, are mostly

subtle, little-discussed rules of judgement and plausible reasoning, acquired only through experience, which are not widely shared nor publicly available. The performance level of an expert system is primarily a function of the size and quality of the knowledge base that it possesses.

An expert system must codify this body of expert knowledge into representations which can be manipulated by the system in order to reason or infer solutions. The representation of the expert knowledge is so critically important that the construction of expert systems often is referred to as knowledge engineering [5]. Several different approaches can be used to represent domain specific facts of the knowledge base. Two methods used successfully are frame-based and rule-based representations.

Frames [10] contain common information associated with the context of a given situation or object. Slots within each frame may store rules, calculation procedures, pointers to other frames, or other information that is relevant to the given situation. Although frames are useful in handling complex situations involving objects and classes of objects, they are infrequently used in most of the current research on expert systems, and not discussed further.

By far the most prominent approach to represent domain specific facts is by production rules [3,4,9]. Production rules are also called if-then or situation-action rules.

Some typical examples of expertise can be expressed with the following rules:

If there is snow on the ground, then it is cold.

If: it looks like a duck, and
it walks like a duck, and
it quacks like a duck,
Then: you've got a duck!

If: the stain of the organism is gram-positive, and
the morphology of the organism is coccus, and
the growth conformation of the organism is clumps
then: (0.7) the identity of the organism is
staphylococcus.

All of the above rules, which may represent expert knowledge, have the following form:

```
IF  (condition 1 is true)
    (condition 2 is true)
    .
    .
THEN (conclusion 1 is true)
      (conclusion 2 is true)
      .
      .
```

In rule-based systems, the knowledge base typically comprises rules of this form. The LISP language is the most common choice for writing expert systems and for coding production rules. Because LISP programs and data are both structured as lists, data structures are easily manipulated by the programmer and by the expert system itself. This flexibility is critical since the size and shape of data structures evolve unpredictably as an expert system runs [7,13].

Please note that there are many ways to represent rules in rule-based systems, implemented with LISP. In every

production rule used by FINPLAN, for example, the atoms RULE, rule numbers, IF, and THEN are superfluous to the data structure, and exist only to make the knowledge base more readable. The IF and THEN clauses of each rule are easily accessed, so that the conditions in each rule can be quickly tested and, if all conditions are satisfied, conclusions can be inferred. Conditions (or conclusions) are tested for a match with already existing facts. FINPLAN, as well as any other expert system, maintains such facts in its global data base.

The global data base

The global data base acts as a working memory by storing input data, system status, or inferred conclusions. All of this information represents facts or assertions about the world. Within a LISP implementation, such facts easily can be represented as lists of atoms. All of them are collected together in a list that becomes the value of the global data base.

FINPLAN initializes its global data base by gathering input data from the user via an interactive interview. It then invokes a function to add each input item to the data base, called FACTS, which simply is a list of all items added.

Other expert systems may employ a global data base with a more complex organization than the one FINPLAN uses. For example, the MYCIN system [11] uses a collection of 4-tuples, consisting of an associative triple and a

certainty factor (CF), which indicates on a scale from -1 to 1 how strongly the fact has been confirmed ($CF > 0$) or disconfirmed ($CF < 0$). Hence,

(IDENTITY ORGANISM-2 KLEBSIELLA .25)

is interpreted as "The identity of organism-2 is Klebsiella with certainty 0.25."

As another example, the DENDRAL system, which performs chemical analysis, uses complex graph structures that represent molecules and molecular fragments as the global data base.

Whatever the organization of the global data base, an important characteristic remains that it is the sole storage medium for all state variables of the system. All information relevant to the solution of the problem and to be recorded must go there. Every rule within the knowledge base has universal access to every fact within the global data base, so that any fact put there is potentially detectable by any rule.

Inference engine

The inference engine, which is the control structure or the heart of an expert system's program, provides much of the variation found among different expert systems. It operates by scanning each rule within the knowledge base until one is found which successfully matches against the current state of the global data base. Its action results in a modified data base, and scanning resumes, either continuing with the next rule or beginning again with the

first. Three methods of rule application are common. Depending on the direction of the solution path, these methods include forward chaining, backward chaining, or a combination of both. The problem's structure and complexity determine which method is most appropriate.

Forward chaining. In forward chained systems, also called data-driven systems, the inference engine starts with a collection of facts and tries all available rules over and over, adding inferred conclusions to the global data base, until no further rules apply. In this case, assertions in the IF clause of each rule are tested for a match with an established fact within the global data base and, if each assertion evaluates to TRUE (i.e. it is part of the data base), the triggered conclusion is also added to the data base. Often, the last deduced inferences comprise the system's output which is the advice ultimately offered to the user. All other items of the global data base represent either input data or intermediate conclusions drawn by inference from the production rules within the knowledge base. Forward chaining is used exclusively by FINPLAN.

Backward chaining. An inference engine performs backward chaining if it starts with an unsubstantiated hypothesis (the goal) and tries to prove it. Because expert systems using backward chaining work from goals to subgoals, they often are called goal-driven systems. The strategy involves finding rules that relate to the goal or

hypothesis and then verifying each of the conditions which demonstrate the hypothesis.

In order to verify conditions, the global data base first is consulted for confirmation of each condition. If unsuccessful, other rules are sought that can verify the condition, which now has become a new hypothesis. In this way, the strategy is recursive, and the program searches backward through its rules until it verifies the hypothesis or cannot go backward any further. At this point the system turns to the user and asks for additional relevant information, which will augment the global data base.

A primary advantage of backward chaining is that information is sought and rules are applied that specifically relate to the overall goal. Unlike forward chained systems, backward chained systems do not appear to work by trial and error, asking apparently unrelated questions. A disadvantage, however, prevents users from volunteering relevant information about the problem. This handicap often makes backward chaining unacceptable when a rapid response is required.

Combination. The combination method, which uses both forward and backward chaining for control, merges advantages of both approaches together. Information is sought and rules are applied that specifically relate to the overall goal, while users also may volunteer relevant information which speeds processing time. Large search spaces, or search spaces that are hierarchically divided,

present the most appropriate opportunities for employing the combination method. Such systems search both from the initial state and from the goal or hypothesis state and utilize a relaxation type approach to match the solutions at an intermediate point.

The justifier

The justifier explains and justifies the program's output. Many AI practitioners consider an explanation facility as an essential component of an expert system because users are not expected to know or understand the whole program. Users seek help from an expert system because they want advice about their problem and will take action based at least partly on that advice. Therefore, due to moral, legal, and ethical issues, users must understand the rational basis for the system's decisions.

HOW and WHY inquires, prompted by the user, depict two common modes of explanation. HOW inquiries ask questions such as "How did you deduce that <conclusion>?" The system responds by returning all facts that allowed inference of the specified conclusion. It is applicable to all three methods of control: forward chaining, backward chaining, and a combination of both. FINPLAN incorporates this type of query in its own explanation facility. WHY inquires propose questions such as "WHY do you need <fact>?" Applicable to systems utilizing backward chaining only, responses to WHY inquiries return the conclusion (hypothesis or goal) dependent upon the specified fact.

Both modes of explanation require maintenance of a continuous record of what data and hypotheses the expert system considers during the course of its execution. An additional system variable records this information by maintaining a list of all rules successfully used. This list is then used to identify the production rule which allowed an inference of the questioned item, so that all facts that allowed its inference may be returned to the user.

The language processor

When an expert system requests input data, accepts data, answers questions, or explains its behavior, communication takes place between it and the user. Language processing often focuses on the pervasive and separate discipline of AI concerning natural languages. For the most part, current expert systems utilize the simplest natural language techniques that support the level of performance required. Most systems use some restricted variant of English, but in some cases systems communicate via a graphics or structure editor. Some language processors parse and interpret user queries, commands, and volunteered information. Conversely, processors also format output information generated by the system. Many systems that use simple techniques exploit canned text to generate messages to the user. FINPLAN is a member of this latter class of systems where fixed messages are selectively chosen for output, and contents of rules are

phrased and expressed in conversational English so that they may be directly outputed to the user.

Conclusion

This chapter summarized what an expert system is and how it works. Basic components of a rule-based expert system, such as FINPLAN, include a knowledge base, global data base, inference engine, justifier, and language processor. The knowledge base stores the knowledge, typically expressed by rules of reasoning. The global data base acts as a working memory by storing established facts and assertions, and holding input data and system status information. The inference engine controls rule interpretation in a forward, backward or combined fashion. The justifier explains the system's behavior and offers credibility for the system in the eyes of the user. Finally, the language processor, in some form or another, controls and processes all interaction between the system and the user.

CHAPTER III OVERVIEW OF FINPLAN

General Comments

System Constraints

This chapter explains the system structure of FINPLAN. FINPLAN's purpose is to demonstrate the methodologies developed for the domain of personal financial planning. From the start, it's development has been shaped by several important constraints.

First, the time permitted for a thesis effort does not allow for the development of a commercially useful, comprehensive and complete expert system. The time construction of early expert systems took 20-50 man-years. Some simple systems have been reported to have been built in as little as 3 man-months, but a complex system still tends to require as long as 10 man-years to complete. Using present techniques, the time for construction of a typical domain specific system appears to be converging towards 5 man-years per system [6].

Second, the assistance of a human expert within the domain of personal financial planning was not available. The quality, or performance, of any expert system depends upon the quality of heuristics and knowledge imparted to its knowledge base by a human expert. Hence, FINPLAN lacks a high degree of performance.

Nevertheless, despite these limitations, FINPLAN exhibits all of the design features and system architectures necessary to produce a potentially useful expert system in the domain of personal financial planning.

Composition of FINPLAN

Because the principal purpose of the thesis effort is to demonstrate that expert system concepts can be applied to the domain of personal financial planning, a completely comprehensive system is not necessary. Two specific tasks of financial planning were chosen as sufficient in order to consider all significant design problems associated with the overall spectrum of financial planning. These two tasks include estimating life insurance needs and making investment portfolio recommendations.

Because the calculation of life insurance needs is numerically oriented, expert systems are not well-suited for it. Nevertheless, FINPLAN uses a methodology (explained below) that was developed specifically to address this class of problems. Other numerically oriented problems, which might be included in a more comprehensive financial planning system, presumably can be based on the same architectures used in this portion of FINPLAN.

Recommendations regarding an investment portfolio comprise the second task performed by FINPLAN. Determining an investment portfolio best suited to the user is perhaps the most important function required in financial planning. This task, unlike the numerically oriented

problems, is better suited to expert system applications. Most of its output is directly inferred through the use of production rules.

Design Goals

A major constraint was the need to design an expert system that runs on a personal computer. FINPLAN was ultimately implemented on a Corona PC with 128K bytes of available memory. This machine environment easily accommodates the present system. An improved version of FINPLAN, with a significantly expanded knowledge base, also is anticipated to run within this environment. This capability was accomplished by subfactoring the overall tasks of FINPLAN into smaller tasks. Many of these tasks in turn are broken down into even smaller subtasks, each requiring only a portion of the overall body of production rules. Hence, the total body of knowledge is factored into many smaller and separate bodies of knowledge bases.

A second goal called for a program that was simple and easy to use. The user may be a novice in either personal computing or financial planning, or both, and still be able to operate FINPLAN easily and effectively. All instructions are self-explanatory. Most user responses require the touch of only a single key, and the variety of possible user responses are kept to a minimum.

A third demand was to keep the system architecture as simple as possible. As a result, production rules in the if-then format are used as the form of knowledge

representation. All rules are encoded directly in the English dialect and stored as such, so that no translation by a language processor is required by the explanation facility when interacting with the user. Also, forward chaining is used exclusively as the direction of search for all inferences.

Finally, another goal was to maximize the speed of FINPLAN's execution. IQLISP, the language in which the system is written, is very conducive to recursive procedures. However, because recursion can be very memory intensive and often slows processing time, iteration in place of recursion is emphasized wherever possible.

Estimating Life Insurance Needs

Introduction

Many tasks included in personal financial planning require numerically oriented computations. Developing a family budget, determining and analyzing net worth, and estimating life insurance needs represent some of these tasks. Because expert systems rely upon inferences, not numerical calculations, such numerically oriented tasks are not well-suited for solution by expert systems.

Nevertheless, such numerically oriented tasks represent integral parts of financial planning, and any comprehensive expert system that addresses the domain of financial planning must accomodate such tasks. Therefore, the estimation of the client's life insurance needs was chosen as a representative task of this class of problems, and a

methodology was developed for FINPLAN to handle it successfully.

Methodology Used

FINPLAN uses two independent processes in order to estimate the life insurance needs of the client-user. One process performs the numerical calculations required to determine the actual amount of life insurance needed, while the second process infers data which can be used to explain the rationale behind the numerical calculations. It is only the second process then, that employs expert system concepts.

An algorithm, prepared by the San Mateo, Calif., financial advisory firm of Bailard, Biehl and Kaiser Inc., serves as the framework for both processes. The algorithm represents a straight-forward method for estimating an adequate dollar amount of insurance to support the client's survivors in the event of his death. Sums to be paid or accumulated at the client's death are first identified. The gap between the survivors' potential income and required living expenses is then calculated. This amount is reduced to its present value in order to allow for the additional money earned by investing the life insurance proceeds. Social Security benefits also are considered in the present value computation. The total insurance need is finally determined by adding the immediate and long-term expenses and subtracting the value of investment assets currently available.

Sample Dialog

Input. In order to begin estimating life insurance needs, FINPLAN gathers basic information about the client in an interactive dialog. For example, a fund should be provided at the time of death in order to cover future college education costs for any children. Information required to estimate the amount of this fund is derived from the following questions (user responses are indicated by preceding double asterisks):

Do you have any children?

**YES

Do you intend to provide a college fund for your children?

**YES

What kind of school do you have in mind?

1 - Private

2 - Public

**1

How many children?

**2

During the course of the above dialog, the following statements are added to the global data base:

YOU INTEND TO PROVIDE A COLLEGE FUND
YOU HAVE A PRIVATE COLLEGE IN MIND

These statements are later used to infer the rationale behind the college fund calculations, which comprise a part of the overall insurance needs, and can be accessed via the explanation facility. Calculation of actual dollar amounts, however, is made numerically using local variables. In determining the college fund requirement, for example, appropriate values are assigned to local variables which represent a predetermined college cost per

child and the number of children. These variables then are used to compute the value of the total college fund requirement, which also is represented by a local variable and which later can be displayed as output. All other computations used to determine the total amount of life insurance needs are made (and explained) in a similar manner.

Output. After all computations and inferences are made, a summary of the results is displayed. Figure 3-1 shows a sample output display.

ESTIMATING YOUR LIFE INSURANCE NEEDS:

1 - Funeral, estate taxes, etc.	\$ 5000
2 - Settle non-mortgage debt (consumer debt)	4500
3 - Emergency fund	5000
4 - College fund	72000
5 - Expected living expenses:	
a - Average annual living expenses	\$ 21000
b - Spouse's average annual income	14000
c - Annual Social Security benefits	5000
d - Net annual living expenses (a-b-c)	2000
e - Investment rate factor	22
f - Total living expenses needed (d x e)	44000

6 - Total monetary needs (1+2+3+4+f)	\$130500
7 - Total investment assets in hand	35000

8 - Life insurance needs (6 - 7)	\$ 95500

Figure 3-1: Sample Life Insurance Output

While viewing the output the user has the option of inquiring about any of the items displayed. The program responds to each inquiry by displaying all the facts which allow the deduction of the questioned item. The explanation facility of FINPLAN does this by scanning a list of all production rules successfully used by the

inference engine, identifying the applicable rule from which the questioned item was deduced, and displaying all facts contained in the IF portion of that rule.

In order to understand how the explanation facility operates, consider first how FINPLAN processes input information. Assume the input information obtained from the above sample dialog regarding estimation of a college fund. Figure 3-2 shows a portion of the life insurance knowledge base which applies to this college fund estimate. After FINPLAN interviews the client regarding a college fund, the global data base will include the following facts:

YOU WANT TO ESTIMATE LIFE INSURANCE NEEDS
THIS COST IS BASED ON AMERICAN COUNCIL ON EDUCATIONS
ESTIMATE
ASSUME A 4 YEAR EDUCATION FOR EACH CHILD
YOU INTEND TO PROVIDE A COLLEGE FUND
YOU HAVE A PRIVATE COLLEGE IN MIND

The system initialized the global data base with the first three facts, which may be considered as given. The last two facts had been established and added to the data base as a result of interviewing the client.

In the next step FINPLAN forward chains the life insurance knowledge base whereby each rule in Figure 3-2 is triggered, inferring additional facts which are added to the global data base. Each rule triggered in this step is added to a separate list structure, RULES-USED, to keep track of all rules successfully used. RULES-USED is needed by the explanation facility. Hence, after the inferences

RULE 1

IF: YOU WANT TO ESTIMATE LIFE INSURANCE NEEDS
THEN: YOU MUST ADD IT IN ORDER TO ESTIMATE TOTAL
MONETARY NEEDS
YOU MUST CONSIDER TOTAL MONETARY NEEDS
YOU MUST CONSIDER IT IN ORDER TO ESTIMATE TOTAL
MONETARY NEEDS
LIFE INSURANCE NEEDS SHOULD BE ESTIMATED
SUBTRACT INVESTMENT ASSETS IN HAND FROM TOTAL
MONETARY NEEDS

RULE 2

IF: YOU MUST ADD IT IN ORDER TO ESTIMATE TOTAL MONETARY
NEEDS
THEN: THIS COST SHOULD BE PROVIDED FOR AT TIME OF DEATH

RULE 3

IF: THIS COST SHOULD BE PROVIDED FOR AT TIME OF DEATH
THEN: YOU MUST INCLUDE FUNERAL EXPENSES IN TOTAL
MONETARY NEEDS
YOU MUST INCLUDE NON-MORTGAGE DEBT IN TOTAL
MONETARY NEEDS
YOU MUST INCLUDE EMERGENCY FUND REQUIREMENT IN
TOTAL MONETARY NEEDS
YOU MUST INCLUDE COLLEGE FUND REQUIREMENT IN TOTAL
MONETARY NEEDS
YOU MUST INCLUDE TOTAL LIVING EXPENSES IN TOTAL
MONETARY NEEDS

RULE 4

IF: THIS COST IS BASED ON AMERICAN COUNCIL ON EDUCATIONS
ESTIMATE
YOU HAVE A PRIVATE COLLEGE IN MIND
THEN: FIGURE \$9000 A YEAR PER CHILD

RULE 8

IF: YOU INTEND TO PROVIDE A COLLEGE FUND
FIGURE \$9000 A YEAR PER CHILD
ASSUME A 4 YEAR EDUCATION FOR EACH CHILD
YOU MUST INCLUDE COLLEGE FUND REQUIREMENT IN TOTAL
MONETARY NEEDS
THEN: THERE IS A NEED FOR PROVIDING A COLLEGE FUND

Figure 3-2: Partial Life Insurance Knowledge Base

are made and before any inquiries are asked, RULES-USED includes all the rules shown in Figure 3-2. In addition, because each rule is enabled, the global data base is now expanded and includes all the facts contained in the THEN portion of each rule of Figure 3-2.

At this time, after viewing the insurance output (shown in Figure 3-1) and selecting the inquiry mode, the client-user may inquire about the college fund computation. Specific code in the program tells the explanation facility, in this case, to look for and explain either the fact "THERE IS A NEED FOR PROVIDING A COLLEGE FUND" or "THERE IS NO NEED FOR PROVIDING A COLLEGE FUND." In this case because the sample dialog indicated a college fund requirement, the former fact will be found in the global data base and RULE 8 identified in RULES-USED. The explanation facility then dissects RULE 8 and displays the following rationale:

THERE IS A NEED FOR PROVIDING A COLLEGE FUND
was deduced from the following:

- 1 - YOU INTEND TO PROVIDE A COLLEGE FUND
- 2 - FIGURE \$9000 A YEAR PER CHILD
- 3 - ASSUME A 4 YEAR EDUCATION FOR EACH CHILD
- 4 - YOU MUST INCLUDE COLLEGE FUND REQUIREMENT IN TOTAL MONETARY NEEDS

Enter line number of item you wish to question
Else enter X

Further queries may be made about any of the displayed items offered as explanation. For example, if the user

now requests information on the amount of money per child (item 2 above), a separate explanation is provided:

FIGURE \$9000 A YEAR PER CHILD
was deduced from the following:

1 - THIS COST IS BASED ON AMERICAN COUNCIL ON EDUCATIONS ESTIMATE

2 - YOU HAVE A PRIVATE COLLEGE IN MIND

Enter line number of item you wish to question
Else enter X

When the end of a line of reasoning is reached, such as when "YOU HAVE A PRIVATE COLLEGE IN MIND" is questioned next, FINPLAN responds with the following message:

YOU HAVE A PRIVATE COLLEGE IN MIND
was either determined in the interview or given.

Strike any key to continue . . .

Once the user chooses to continue the program the insurance summary is displayed again, at which time the user may request additional explanations of any item or continue with the investment portion of the program.

Making Investment Portfolio Recommendations

Introduction

Suggesting recommendations regarding an investment portfolio best-suited to the situation and needs of a client-user represents a major task of any financial planning system. It also represents a task in financial planning which is most conducive to expert system concepts. Recommendations regarding an investment portfolio made by FINPLAN include:

- a) an appropriate number of investment classes, ranging from one to four classes. This number is based on the client's need for diversification.
- b) amount of capital to be invested in each investment class. This amount is based on the total capital available and the number of classes FINPLAN selects.
- c) identification of each investment class. The selection of classes is based on the best match between the client's goals or constraints and the degree to which each possible investment class satisfies those goals or constraints.

Methodology Used

Determining the client's goals. Constraints/goals are inferred through a specific body of production rules starting with basic information about the client-user. This basic information is obtained as input from an interactive interview. The client is characterized with respect to each of seven specific constraints/goals, which later are used to identify the most appropriate investment classes for the client. Generally, there are at least three possible characterizations for each constraint. An example of a complete set of constraint characterizations for a given client follows:

- 1 - Objective should be retirement security
- 2 - Financial risk should be low
- 3 - Liquidity should be moderate
- 4 - Tax exemption is moderately important
- 5 - Inflation protection should be high
- 6 - Effort required should be low
- 7 - Skill required should not be demanding

Determining degree of diversification. FINPLAN infers the degree of diversification needed by the client using another separate body of production rules. The degree of

diversification determines the number of investment classes ultimately recommended. All final inferred conclusions regarding diversification can be traced to:

- a) the recommended riskiness of the investment portfolio, which is based on the client's risk tolerance and major investment objective.
- b) the amount of capital the client has available for investment.
- c) the degree of desire the client indicates for participating in the management of his portfolio.

Amount of capital. After an appropriate number of investment classes is inferred from the degree of diversification, the amount of capital to be invested in each class is numerically calculated. FINPLAN still makes inferences based on production rules, however, in order to provide a means of explanation. The computed amount of capital is truncated to the next lowest \$1000 unit, and any remaining dollars are recommended to be placed in the client's savings account.

Investment selection algorithm. Selection of the most appropriate investment classes is made by evaluating and ranking each investment class according to how well each satisfies the client's desired constraints/goals. The evaluation of each class is done numerically, but inference procedures still exist to be used by the explanation facility. The selection algorithm that is applied to each investment class is presented in Figure 3-3.

```
FOR each investment class i (i = 1,2 . . 25) DO
    RANKi = 0
    FOR each of the 7 constraints (j = 1,2 . . 7) DO
        1. SUM = INV(i,j) - GOAL(j)
        2. IF SUM > 0 THEN SUM = 5 * SUM
        3. RANKi = RANKi + abs(SUM)
    UNTIL all 7iconstraints have been considered
    assign RANKi to investment class i
UNTIL all investment classes have been considered
```

Figure 3-3: Selection Investment Algorithm

INV(i,j) used in the algorithm and whose values appear in Table 3-1 represents the characterization of investment class i with respect to constraint j. GOAL(j) indicates the client's characterization with respect to constraint j.

Note that the class with the lowest RANK value represents the best match for the client's given set of constraints/goals. A RANK = 0 for investment class i indicates a "perfect" match for investment class i.

Each time a constraint or goal i is not satisfied, the rank for that investment class is increased by at least the difference between INV(i,j) and GOAL(j). Sometimes, however, this penalty is increased by a multiple of 5 in step 2 of the algorithm. For example, assume the client wants an investment class where "financial risk can be moderate" (GOAL(1) = 20). A class whose financial risk is characterized as moderate (INV(i,j) = 20) would receive a contribution of the quantity of zero added to its RANK (a perfect match, as far as risk is concerned). If its risk were low (INV(i,j) = 10), a 10 point penalty would be added to its RANK. On the other hand, if it were a high risk

TABLE 3-1: CONSTRAINTS USED IN THE
INVESTMENT SELECTION ALGORITHM

<u>i</u>	<u>CONSTRAINT/GOAL</u>	<u>CHARACTERIZATION</u>	<u>INV(i,j) OR GOAL(j)</u>
1	Financial risk	Low	10
		Moderate	20
		High	30
2	Objective	Income	5
		Retirement security	10
		Growth	15
		Speculation	20
3	Tax exemption	High	4
		Moderately important	8
		Not important	12
4	Inflation protection	High	3
		Moderately important	6
		Not a consideration	9
5	Liquidity	High	2
		Moderate	4
		Not a consideration	6
6	Effort required	Low	2
		Moderate	4
		High	6
7	Skill required	Not demanding	2
		Can be moderate	4
		Can be high	6

investment class ($INV(i,j) = 30$), a 50 point penalty would be added to its RANK. This additional penalty is assigned in order to compensate for the presumed fact that a moderate risk-taker would be far more adverse to accepting a high risk investment than he would be to accepting a low risk investment.

All values associated with each type of constraint/goal ($INV(i,j)$ and $GOAL(j)$) are ordered in a predetermined way such that the additional penalty is invoked only when an attempted match "misses" on the non-conservative side (i.e. as in the risk example above).

The values associated with each constraint/goal relative to other constraints vary in magnitude. Hence, each constraint category may be weighted and its relative proportion within the algorithm assigned. Thus, risk considerations may be made more important than tax considerations, which may be made more important than required effort, etc.

Investment output. Once all possible investment classes are ranked using the investment selection algorithm, a desired number of classes are identified and each displayed in ascending order of its RANK (lowest RANK first). Figure 3-4 shows a sample output. The user has the same option to inquire about any of the numbered items displayed, as he could during the insurance portion of the program. The explanation facility works exactly the same way as described earlier.

You have \$ 56000 available for investment. Based upon your situation, temperament, and goals, FINPLAN recommends that your portfolio includes investments allocated to the investment classes listed below.

- 1 - 3 investment classes are recommended.
- 2 - \$ 18000 is recommended for investment in each class.
- 3 - Class 1: \$ 18000 in ESTABLISHED DIVIDEND STOCKS
- 4 - Class 2: \$ 18000 in INCOME-ORIENTED MUTUAL FUNDS
- 5 - Class 3: \$ 18000 in HIGH GRADE MUNICIPALS

Place the remaining \$ 2000 in your savings.

Figure 3-4: Sample Output of Investment Recommendations

CHAPTER IV SPECIAL DESIGN CONSIDERATIONS AND ALTERNATIVES

Introduction

During the course of developing FINPLAN, many design problems unique to either expert systems or to the domain of financial planning were encountered. Each of these problems could have been solved by any one of several alternatives. For each design problem considered in this chapter, the problem is first defined, the chosen solution is described, and alternative solutions are then noted and briefly discussed. For a discussion of additional variations of alternative architectures common to expert systems in general see [8,12].

Design Considerations and Alternatives

Controlling the Size of the Knowledge Base

The problem. A method must be used to restrict the size of the knowledge base, keeping the number of production rules within a manageable limit. The growth potential of a knowledge base may best be explained with an example. Mutually exclusive characterizations for the client regarding his need for liquidity, tax exemption, inflation protection and other investment constraints are inferred from information about the client. The need for liquidity in any investment, for example, could be inferred

exclusively from the client's level of emergency reserves: the lower his level of reserves, the greater his need for liquidity. Because each of three possible values which represent the level of reserves could directly infer one of the three possible values representing the need for liquidity, a total of three production rules would be adequate to infer an appropriate value for liquidity. If, however, the client's degree of risk tolerance, in addition to his level of reserves influences the need for liquidity, more production rules would be needed. Since his risk tolerance is characterized by one of three possible values, a total of 9 potential sets of conditions now must be considered in order to infer the appropriate liquidity need. Introducing a third factor, which has four possible values, raises the number of production rules needed to 36. Clearly, a method must be used that eliminates the need for a comprehensive set of production rules covering all possible discreet sets of multiple conditions, in order to keep the number of rules from becoming prohibitively large.

Solution chosen. In most situations involving multiple factors, a single factor is dominant when inferring a conclusion. The need for liquidity, for example, is determined primarily by the level of emergency reserves. In a few cases, however, unique combinations of other secondary factors represent exceptions, whereby a conclusion differs from one inferred by the consideration

of emergency reserves alone. For example, liquidity should be moderate if reserves are moderate unless income is secure and the client is an aggressive risk-taker, in which case liquidity can be low. Rather than incorporating rules that cover all possible combinations of values for all input factors, FINPLAN identifies rules which produce only those conclusions that differ from what the primary factor normally produces. Such cases are labeled exceptional cases, and the rules that address them are called exception rules. Rules whose inferences are based solely on the primary factor are called default rules. Together, exception and default rules cover an exhaustive listing of potential inferences without including an exhaustive listing of potential rules.

FINPLAN isolates these two types of rules into separate knowledge bases. First, the exception rules are forward chained. If no conclusion is successfully inferred for a specific constraint (ie. liquidity, risk, skill, etc.), then the default rules for that category are forward chained. In this way, a single mutually exclusive conclusion for each constraint category is guaranteed without considering a comprehensive set of rules covering every possible set of conditions. Only the exception and default rules need to be considered.

Alternative solutions. An alternative solution also involves the identification of exception and default rules, but this time includes them together in a common knowledge

base. Rules within this knowledge base are ordered so that the exception rules follow the default rules. This methodology, however, will infer two different characterizations for the same constraint category whenever an exception rule is triggered. To correct this result second inferences must be identified, and, if one occurs, all effects from the first inference must be undone. This process includes removing the previously inferred fact from the global data base as well as the previous rule from RULES-USED, the list containing all successfully used rules. FINPLAN initially implemented this methodology, but every inference for each constraint category had to be checked for a previous inference, resulting in a slower processing time. In addition, the code was awkward and required dangerous IQLISP functions for list surgery.

Another potential solution, when inferring constraint categories, identifies default and exception rules. This solution also includes both types of rules in a common knowledge base with exception rules ordered last. This time, however, when two or more inferences are made concerning the same category, no attempt is made to remove the inappropriate items from the global data base and RULES-USED. Because items are added to the front of the global data base and RULES-USED, the last items added will always be the first items on both data structures. Hence, because exceptions are triggered last, the relevant items appear first on both the global data base and RULES-USED.

As long as any procedure that accesses these data structures searches these lists from front to back and stops scanning whenever the first item for a desired category is found, the effects of inappropriate inferences simply can be ignored.

This design carries some disadvantages with it, however. First, the global data base and RULES-USED data structures become larger than necessary. The unnecessary growth of data structures is particularly harmful because the conservation of memory usage is critical in any expert system implemented on a personal computer. Furthermore, this design is philosophically undesirable because contamination of the global data base and RULES-USED should not be condoned. Special action must be taken in order to avoid erroneous output resulting from their contamination. This approach circumvents the intended content of the global data base and RULES-USED where any further access or manipulation of these data structures must be done with extreme caution.

Numerically Oriented Tasks

The problem. Expert systems are best suited for qualitative tasks, such as in the case of deductive reasoning, not quantitative tasks. Unfortunately, not all tasks in financial planning involve qualitative advice inferred from lines of deductive reasoning. In those tasks that require numerical procedures, some kind of quantitative method using numerical computations must be implemented. Several methodologies are available.

Solution chosen. FINPLAN uses two independent processes, one to infer lines of deductive reasoning and the other to make traditional arithmetic calculations, whenever quantitative computations are required. The insurance portion of FINPLAN depicts an example of this methodology, which was discussed in Chapter III.

Alternative solutions. Rather than storing numerical values in local variables and manipulating these variables in an independent process, the values instead could be stored directly in each appropriate fact within the global data base. These stored values then could be extracted, manipulated by traditional methods of calculation, and the calculated value returned to the appropriate fact within the global data base. For example, total living expenses in the insurance portion of FINPLAN is determined by multiplying the net annual living expenses by an investment rate factor. Each of these values may be stored in the global data base as follows:

(YOUR NET LIVING EXPENSES HAVE BEEN DETERMINED TO BE <value-1>)

(YOUR INVESTMENT RATE FACTOR IS <value-2>)

(TOTAL LIVING EXPENSES HAVE BEEN COMPUTED TO BE <value-3>)

When the computation of total living expenses is to be made, <value-1> and <value-2> can be extracted from their respective facts within the global data base and used to compute total living expenses. This new value then

replaces <value-3> which is returned to and stored in the global data base.

FINPLAN currently stores values used in the investment selection algorithm for the evaluation and ranking of investments using this general method. An advantage includes storage efficiency, where values can be directly linked to appropriate facts within the global data base without the use of local variables. Accessing these values, however, may represent a disadvantage since the fact associated with the value first must be retrieved from the global data base before the value can be extracted.

Selection of Investment Classes

The problem. Investment classes that best match the client's goals and constraints must be selected. Several alternatives exist.

Solution chosen. FINPLAN implemented a numerical algorithm that ranks every investment class included within the knowledge base. Inferences had to be forfeited in this evaluation process because no methodology could be developed which used inferences that satisfactorily ranked all investment classes. Chapter III discussed the investment selection algorithm in detail.

Alternative solutions. Before the selection algorithm was elected, a methodology which strictly used inferences by production rules was attempted. This design, however, proved to be unsatisfactory.

The basic concept on which this design was based matches the client's goals with each investment's characteristics. For example, if the client is determined to be a moderate risk-taker, his goal for risk is represented by the following fact: FINANCIAL RISK SHOULD BE MODERATE. If the client accepts an investment class with moderate risk, then he presumably would also accept an investment with low risk. Therefore, production rules translate his original goal into the following constraints (to be tested against the investment's characteristics for risk):

FINANCIAL RISK CAN BE LOW

FINANCIAL RISK CAN BE MODERATE

In this manner, production rules expand all client goals so that they describe all acceptable characterization values. Another set of production rules are then forward chained in order to test for exact matches between each investment's characteristics and the client's acceptable goals. An investment class is inferred only when each characteristic for it exactly matches every client goal.

The lack of control over the number of investment classes selected represents the primary disadvantage of this design. In some cases no investment classes match the client's profile of goals. On the other hand, if a client's profile is fairly liberal, many investment classes fit his goals. Hence, selection of a specified number of classes can not be easily accomplished. An attempt was

made to iterate the deductive process, accepting only one investment class at each pass, until the appropriate number of investment classes was inferred. Nevertheless, this attempt failed because of the inability to rank investments when multiple classes were inferred during a single iteration. Therefore, for these reasons the election of investment classes by strict inference procedures was abandoned in favor of using a numerical algorithm.

Storage of Numerical Values

The problem. As noted earlier, numerical values used in the investment selection algorithm are stored as a part of the production rules within the knowledge base itself. Even though investment classes are not directly inferred, production rules still exist that can be used to explain why a particular investment was selected. Since a rule exists for each investment class, these rules provide a convenient storage medium for the evaluation score computed for each investment class. Rules inferring client constraints also represent a convenient medium for storing the numerical values associated with each client constraint. These values can be stored in a variety of ways.

Solution chosen. The design implemented in FINPLAN stores these values as dotted pairs. For example, an inference for blue chip stocks, with an evaluated score of 22, would be represented as: (BLUE CHIP STOCKS . 22).

The utilization of dotted pairs is memory efficient. The storage of the above representation consumes three cons-cells; any other representation would consume at least four cells. Also, this representation provides a convenient means to print the contents of an assertion without displaying the numerical value itself. This advantage is useful for execution of the explanation facility, and is accomplished by using the IQLISP function MAPCAR. MAPCAR and its related IQLISP functions apply themselves to the CAR's of successive tails of their list arguments, thereby ignoring the CDR of each argument's last cons-cell. Any other representation would have required additional code, with correspondingly slower processing time, in order to preclude the stored values from being printed. Furthermore, each value stored as a dotted pair still is easily accessed by retrieving the CDR of the last element within the assertion. Each assertion (fact) is represented by a list of elements, and IQLISP conveniently provides a function for directly retrieving the last element of any list.

Alternative solutions. Numerical values used by the investment selection algorithm also could be stored in each clause as a separate element. The last element is preferable due to its ease of access. Nevertheless, an extra cons-cell would be required for each value, and special code would be required to print the contents of each clause without displaying the value itself. Another

alternative calls for representing each value by a separate list within the clause. Again, however, more memory and special considerations for printing would be required.

CHAPTER V SUGGESTED IMPROVEMENTS FOR FINPLAN

Introduction

Because the development of FINPLAN represents the first effort to program an expert system handling personal financial planning, a perfected and comprehensive system was never expected. Furthermore, the time permitted for a thesis development and the unavailability of aid from a human expert in the financial field also precluded the hope for a complete and commercially usable system. Nevertheless, a hope is that another person or student, sometime in the future, will take the current status of this effort and advance it several steps further. For this reason, and in order to complete a fair assessment of FINPLAN, the following paragraphs in this chapter identify and briefly discuss limitations and suggested improvements relating to the current system.

Limitations and Improvements

More Tasks

FINPLAN should address additional tasks associated with personal financial planning. Besides inferring recommendations regarding an investment portfolio, the current system addresses only the amount of life insurance needs as a representative task for the class of numerically

oriented problems. Financial planning also encompasses many other tasks within this class. The determination of net worth and the development of a cash budget may represent further relevant tasks for financial planning. Additional forms of insurance, such as health and homeowner's coverage, also may be included in a more comprehensive system. Not only determining adequate amounts of insurance coverage is an important consideration, as FINPLAN currently does for life insurance, but providing advice related to alternative types of insurance plans and the amount of deductibles could also be covered. Additionally, developing financing plans for real estate, including the client's own home, could fall within the purview of a more comprehensive financial planning expert system.

Aspects of financial planning are wide and varied, and not all pertinent tasks within this domain are mentioned here. Nevertheless, the fact remains that FINPLAN must be expanded to cover additional tasks, if it is to evolve into a comprehensive, complete, and commercially usable expert system for financial planning purposes.

Expansion of Production Rules

The lines of reasoning currently incorporated in FINPLAN are often sparse and sometimes show trivial logic. This result is a reflection of the quality and depth of the production rules themselves. It should be noted, however, that the primary purpose of FINPLAN is not to embody

heuristic knowledge expressing subtle, detailed rules known only to a handful of human experts. Rather, its purpose is to demonstrate the feasibility of implementing expert system concepts in the domain of financial planning. Therefore, design features represent the primary focus, and the present absence of high performance can be tolerated.

If, however, FINPLAN evolves into a more comprehensive and useful system, its performance must be upgraded by improving the quality and scope of its production rules. Only a human expert can impart the type of knowledge required for this enhancement. In addition to expanding the knowledge base, a human expert also may recommend improvements in the basic approach taken by FINPLAN in solving some of its tasks. For example, in the investment portion of the program, an expert may recommend that either more or different types of constraints be considered than the seven constraints FINPLAN currently uses.

Better Input Procedures

In many cases FINPLAN allows the client to dictate his own desired constraint characterizations during the course of the interactive dialog, even when the client may not have enough competence to respond wisely. For example, in the investment portion of the program, FINPLAN blindly accepts one of four primary investment objectives as indicated by the client. In the event the client is not sure of which objective would be best suited for him, no provisions exist which could make further inquiries and

offer guidance to the client for selecting the optimal choice. Situations of this kind offer excellent opportunities to employ expert system concepts in an environment where such concepts are most appropriate. This approach is illustrated to some extent by inferences made by FINPLAN in determining the client's need for liquidity and skill level, but additional heuristics should be included to infer other types of input data during the course of the interactive dialog. Again, however, a qualified human expert is needed in order to implement this kind of improved input procedures.

A Self-help Facility

The current system makes no provision for a self-help facility. Some type of a help facility may be useful where detailed definitions of financial terms or explanations of concepts could be offered on demand.

Recursive Explanation Facility

Currently, FINPLAN provides an explanation facility that allows a single line of reasoning to be traced through as many levels of inferences as necessary in order to reach its origin. The explanation procedure is executed iteratively. A drawback of iteration is that only one item, displayed at any given level of explanation, can be traced backward to find the reason for its inference. In order to return to the same level of explanation so that a second or third item also can be traced backwards, the user must first return to the top level and work his way back to

the level in question. Implementation of a recursive explanation procedure would resolve this inconvenience. It should be noted, however, that in order to operate effectively a recursive explanation facility would require a more complicated set of instructions than the user faces currently. Nevertheless, the adoption of such a procedure should be considered in the development of a more complete and usable expert system.

Internal Representation of Knowledge

Many sophisticated expert systems encode rules of knowledge in rigorous mathematical notation. FINPLAN does not take this approach, and uses a simpler design which encodes all assertions and rules directly in English. The following example illustrates a more sophisticated, yet typical representation scheme used in many other expert systems:

English version -

```
(IF ((YOUR PORTFOLIO SHOULD BE AGGRESSIVE)
     (DESIRE TO PARTICIPATE IS HIGH)
     (AVAILABLE CAPITAL IS $12000 OR MORE))
  (THEN ((DIVERSIFICATION CAN BE HIGH))))
```

Internal code -

```
PREMISE: ($AND (SAME PORTFOLIO SHOULD-BE AGGRESSIVE)
           (SAME DESIRE IS HIGH)
           (SAME CAPITAL GTE 12000))
ACTION:   (CONCLUDE DIVERSIFICATION CAN-BE HIGH)
```

The premise of the internal code represents a Boolean combination of one or more clauses (in this case the Boolean function AND identified by \$). Each clause forms a predicate function with an associative triple (object, attribute, value) as its argument. In the first clause

above, the predicate function is SAME, followed by the object PORTFOLIO, attribute SHOULD-BE and the value AGGRESSIVE.

Utilization of internal code similar to the above representation scheme allows efficient and flexible manipulation of the knowledge base. For example, the Boolean function OR could also be used in a production rule in addition to AND, to which FINPLAN is currently restricted. A special language processor is needed when internal representation is used, however, so that the encoded information can be translated into a more readable English-like version for any interaction with the user.

LIST OF REFERENCES

1. Buchanan, Bruce G., "Research on Expert Systems," REP HPP-81-1, Computer Science Dept., Standford University, 1981.
2. Davis, Randall, "Interactive Transfer of Expertise: Acquisition of New Inference Rules," Proc. of IJCAI 5, Vol. 1, pp. 321-328, 1977.
3. Davis, Randall, B. Buchanan, E. Shortliffe, "Production Rules as a Representation For a Knowledge-Based Consultation Program," Artifical Intelligence, Vol. 8, No. 1, pp. 15-45, 1977.
4. Davis, Randall, J. King, "An Overview of Production Systems," in Machine Intelligence 8, E. W. Elcock, D. Michie (Eds.), Ellis Horwood Ltd., Coll House, England, pp. 300-328, 1977.
5. Feigenbaum, Edward A., "The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering," Proc. of IJCAI 5, Vol. 2, pp. 1014-1028, 1977.
6. Gevarter, William S., "An Overview of Expert Systems," REP NBSIR 82-2505, National Bureau of Standards, U. S. Dept. of Commerce, 1982.
7. Ham, Michael, "Playing By the Rules," PC World, Vol. 2, No. 1, pp. 34-41, Jan. 1984.
8. Hayes-Roth, F., D. Waterman, D. Lenat (Eds.), Building Expert Systems, Addison-Wesley Publishing, Inc., Reading, Mass., 1983.
9. McDermott, John, "R1: A Rule-Based Configurer of Computer Systems," Artificial Intelligence, Vol. 19, pp. 39-88, Sept. 1982.
10. Rich, Elaine, Artificial Intelligence, McGraw-Hill, Inc., New York, 1983.
11. Shortliffe, Edward H., Computer-Based Medical Consultations: MYCIN, American Elsevier, New York, 1976.

12. Stefik, M., J. Aikens, R. Balzer, J. Benoit, L. Birnbaum, F. Hayes-Roth, E. Sacerdoti, "The Organization of Expert Systems, A Tutorial," Artificial Intelligence, Vol. 18, pp. 135-173, Mar. 1982.
13. Winston, Patrick H., B. Horn, LISP, Addison-Wesley, Inc., Reading, Mass., 1981.

APPENDIX A LIST OF FILE CONTENTS

The source code listings are segregated into multiple files, where each file contains several functions. The following table lists the name (and type) of each file in alphabetical order, and all functions contained in each file.

TABLE A-1: FILE CONTENTS

<u>FILE</u>	<u>FUNCTIONS</u>	<u>FILE</u>	<u>FUNCTIONS</u>
CONTROL.LSP	MEMBEROF RECALL TESTIF ADDTHEN TRYRULE FORWARDCHAIN PR PR-TEXT PR-SCREEN HOW THENP INQUIRE FIND MATCH ADD ANS	DO-INV.LSP	DO-INVESTMENTS SHOW-CONSTR INQUIRE-CONSTR SHOW-INV INQUIRE-INV INIT.LSP INS.LSP INTRO.LSP
CONV.LSP	CONVERT- CONSTRAINTS	INVEST.LSP	INITIALIZE INS-INTERVIEW INTRO SEL CONT DESCRIP INVESTMENT- RULELIST
C-RULE.LSP	CONSTRAINT- RULELIST	I-PAR.LSP	INV-PARAM INS-RULELIST LOAD
DO-INS.LSP	DO-INSURANCE SHOW-INS INQUIRE-INS	I-RULE.LSP LOAD.LSP QUERY.LSP RANK.LSP RUN.LSP	INV-INTERVIEW RANK-INV EVAL-INV SORT SORT-INVS RUN WAIT STOP-WAIT CHOICE

APPENDIX B FUNCTION DESCRIPTIONS

Descriptions of all source code functions, arranged in alphabetical order, are listed below.

ADD: This function adds the clause represented by NEW to FACTS (the global data base) unless NEW is already in FACTS. If NEW is already present, NIL is returned.

ADDTHEN: This function adds each THEN clause of RULE, if not already on FACTS list, to FACTS list.

ANS: This function helps to print alternative responses to questions asked the user. The numeric NO is printed, followed by the expression PHRASE.

CHOICE: This function defines a message displayed with intermediate and final output in both the insurance and investment portions of the program. The user is given a choice to either inquire about the current output or continue with the program. If the user chooses to inquire about the output, this function calls an appropriate function that controls the explanation facility.

CONSTRAINT-RULELIST: This function infers a characterization value for each of seven user constraints later used to select investment classes. Three separate knowledge bases are defined, inferences are made, and numerical values are assigned to be used by the investment selection algorithm.

CONT: This function displays a message and allows the system to wait until the user responds by striking any key.

CONVERT-CONSTRAINTS: This function defines and infers a knowledge base used to convert established user constraint characterizations into terms used in the investment selection knowledge base (INVESTMENT-RULES). This conversion is necessary because inferences are based on exact matches. In addition, this conversion provides a more appropriate phraseology for the explanation facility.

DESCRIP: This function displays a description of FINPLAN on the screen for the user to review, at his option.

DO-INSURANCE: This function controls the entire insurance segment of the program. It loads all appropriate files and calls all functions needed to conduct the interview, perform calculations, make inferences, and provide a framework for executing the explanation facility.

DO-INVESTMENT: This function controls the entire investment segment of the program. It loads all appropriate files and calls all functions needed to conduct the interview, make inferences, rank investment classes, and show results. It also provides a framework for executing the explanation facility.

EVAL-INV: This function applies the investment selection algorithm to the investment class contained in RULE.

FIND: This function returns the first clause (a fact) from the global data base that matches the pattern specified by PATTERN. PATTERN is comprised of a partial clause with any combination of words missing. If PATTERN cannot be matched in the global data base, NIL is returned.

FORWARDCHAIN: This function makes all possible inferences in a knowledge base by applying TRYRULE to each rule in RULELIST.

HOW: This function represents the primary procedure used by the explanation facility. It searches RULES-USED to find the rule which inferred FACT and, if the appropriate rule is found, returns each clause in the IF portion of the rule. If the rule is not found, appropriate messages are displayed depending on whether FACT exists in the global data base.

INITIALIZE: This function defines the windows used throughout the program, and initializes global variables.

INQUIRE: This function controls the processing of the explanation facility. It processes user input and conditionally executes the HOW function with an appropriate argument so the desired item is explained.

INQUIRE-CONST: This function provides a framework for executing the explanation facility for inquiries about the user's investment constraints. It conditionally calls the HOW function with an appropriate argument so that the desired constraint is explained.

INQUIRE-INS: This function provides a framework for executing the explanation facility for insurance inquiries. It conditionally calls the HOW function with an appropriate argument so that the desired item is explained.

INQUIRE-INV: This function provides a framework for executing the explanation facility for inquiries about the final investment recommendations. It conditionally calls the HOW function with an appropriate argument so that the desired recommendation is explained.

INS-INTERVIEW: This function interactively conducts an interview with the user in order to gather basic information relating to the user's life insurance needs. Local variables used in the insurance algorithm also are assigned appropriate values, and the algorithm itself is executed.

INS-RULELIST: This function defines the body of production rules used in the insurance segment of the program. Initial facts are added to the global data base, and all rules of INS-RULES are forward chained in order to make all possible inferences. (These inferences are used predominantly to explain the insurance algorithm.)

INTRO: This function defines the graphics (composed of windows) displayed at system start-up. It conditionally calls a function which describes FINPLAN or immediately begins a main program segment, depending on the wishes of the user.

INVESTMENT-RULELIST: This function defines the knowledge base containing all investment classes that are considered in the investment segment of the program. Each rule is not implemented as a production rule, and no inferences are made. Instead, each rule is used in the investment selection algorithm and by the explanation facility.

INV-INTERVIEW: This function interactively conducts an interview with the user in order to gather basic information used in the investment segment of the program. Responses are added to the global data base so that future inferences can be made.

INV-PARAM: This function defines three separate knowledge bases and makes all possible inferences from them in order to determine the appropriate degree of diversification for the user's investment portfolio. Dollar amounts for investments also are numerically calculated by this function.

LOAD: This function loads the file specified by FILE into the IQLISP environment.

MATCH: This function is used by FIND to test whether the pattern represented by PAT matches the clause ASSERT, which represents a fact within the global data base.

MEMBEROF: This function returns T (TRUE) if X is a member of the list Y.

PR: This function prints the expression EXP1 immediately followed by the literal LIT on the same line in window WIND.

PR-SCREEN: This function prints the current value of NUMB (the next ordered line number used in the explanation facility), followed by "-", followed by the contents of the clause A. NUMB is incremented for the next time this function is called.

PR-TEXT: This function prints each element within the list EXP on the same line. It is used to print the textual contents of a clause contained in a rule.

RANK-INV: This function evaluates each investment class included in INVESTMENT-RULELIST in accordance with the investment selection algorithm, and records the evaluated score for each investment in INVESTMENT-RULELIST. It performs the evaluations by calling EVAL-INV.

RECALL: This function returns T (TRUE) if FACT is a member of the global data base (FACTS), else it returns NIL.

RUN: The RUN function represents the main body of the program. It initializes and executes each of the main segments of the program: the introduction, the insurance segment, and the investment segment.

SEL: This function displays a message on the screen, providing instructions to the user. A numeric, NO, is printed, followed by PHRASE on the same line in the window specified by WIND.

SHOW-CONSTR: This function displays the intermediate inferences characterizing the user's profile of investment constraints. These constraints are later used to rank investment classes.

SHOW-INS: This function displays the insurance output.

SHOW-INV: This function displays the final investment recommendations.

SORT: This function scans all investments in INVESTMENT-RULES and sets a pointer PTR to the investment with the lowest (highest ranked) evaluation score. The score of the chosen investment is then changed to the exceedingly large quantity of 500 so that a subsequent execution of this function will select another investment with the next lowest evaluation score. Also, the rule containing the chosen investment is added to RULES-USED.

SORT-INVS: This function applies SORT four times so that pointers are set to the four investment classes with the four lowest (highest ranked) evaluation scores.

STOP-WAIT: This function clears the wait message (generated by WAIT) from the screen.

TESTIF: This function returns T (TRUE) if all IF clauses of RULE are part of the global data base, else it returns NIL.

THENP: This function returns T (TRUE) if FACT is present in the THEN portion of RULE.

TRYRULE: This function tests whether RULE is applicable by calling TESTIF. If RULE applies, it attempts to add each THEN clause to the FACTS list by calling ADDTHEN, and returns T.

WAIT: This function defines a wait message which is displayed on the screen as the program performs internal processing.

APPENDIX C
SOURCE CODE LISTINGS

FILE: CONTROL.LSP

```
(DEF 'MEMBEROF
  '(LAMBDA (X Y)
    (SUBSET '(LAMBDA (Z) (EQUAL X Z)) Y)))

(DEF 'RECALL
  '(LAMBDA (FACT)
    (COND
      ((MEMBEROF FACT FACTS)
       FACT)
      [T
       NIL])))

(DEF 'TESTIF
  '(LAMBDA (RULE)
    (PROG (IFS)
      (SETQ IFS
            (CDR (CADR RULE)))
      LOOP
      (COND
        ((NULL IFS)
         (RETURN T))
        ((RECALL (CAR IFS))
         )
        [T
         (RETURN NIL)])
      (SETQ IFS
            (CDR IFS))
      (GO LOOP)))))

(DEF 'ADDOTHEN
  '(LAMBDA (RULE)
    (PROG (THENS SUCCESS)
      (SETQ THENS
            (CDR (CAR (CDDR RULE)))))
      LOOP
      (COND
        ((NULL THENS)
         (RETURN SUCCESS))
        ((ADD (CAR THENS))
         (SETQ SUCCESS T)))
      (SETQ THENS
            (CDR THENS))
      (GO LOOP)))))

(DEF 'TRYRULE
  '(LAMBDA (RULE)
    (COND
      ((TESTIF RULE)
       (ADDOTHEN RULE)
       (SETQ RULES-USED
             (CONS RULE RULES-USED)))
       T)))
    )

(DEF 'FORWARDCHAIN
  '(LAMBDA (RULELIST)
    (PROG (RULES)
      (SETQ RULES RULELIST)))
```

```

LOOP
  (COND
    ((NULL RULES)
     (RETURN NIL))
    ((TRYRULE (CAR RULES))
     )))
  (SETQ RULES
    (CDR RULES))
  (GO LOOP)))))

(DEF 'PR
  '(LAMBDA (EXPI LIT WIND)
    (PRINTC " " WIND)(PRINC EXPI WIND)
    (PRINC LIT WIND)))

(DEF 'PR-TEXT
  '(LAMBDA (EXP)
    (MAPC '(LAMBDA (X)
      (PRINC X WIN)(PRINC " " WIN)) EXP)))

(DEF 'PR-SCREEN
  '(LAMBDA (A)
    (PR NUMB " - " WIN)(PR-TEXT A)
    (COND
      ((EQUAL NUMB 1)(SETQ ONE A))
      ((EQUAL NUMB 2)(SETQ TWO A))
      ((EQUAL NUMB 3)(SETQ THR A))
      ((EQUAL NUMB 4)(SETQ FOU A))
      ((EQUAL NUMB 5)(SETQ FIV A))
      ((EQUAL NUMB 6)(SETQ SIX A))
      ((EQUAL NUMB 7)(SETQ SEV A))
      ((EQUAL NUMB 8)(SETQ EIG A))
      ((EQUAL NUMB 9)(SETQ NIN A)))
    (SETQ NO-REA (ADD1 NO-REA))
    (SETQ NUMB (ADD1 NUMB)))))

(DEF 'HOW
  '(LAMBDA (FACT)
    (PROG (POSSIBILITIES SUCCESS WIN)
      (SETQ POSSIBILITIES RULES-USED)
      (SETQ WIN W4)(WINCLR W4)
      LOOP
        (COND
          ((NULL POSSIBILITIES)
           (COND
             (SUCCESS
              (RETURN T))
             ((RECALL FACT)
              (PRINTC " " WIN)(PR-TEXT FACT)
              (PRINTC " was either determined in the interview or "
                     " given." W4)
              (WINCLR W4)(PRINTC " " W4)
              (PRINTC " Strike any key to continue . . ." W4)
              (CLRBFI)(SETQ X (TYI W4))(SETQ STATUS 'LDR-DONE)
              (RETURN T))
            ST
              (PRINTC " " WIN)(PR-TEXT FACT)
              (PRINTC " is not an established constraint for your sit-
uation. Still, " W4)
              (PRINTC " this investment class is best for you based on
your other " W4)
              (PRINTC " constraints." W4))))
```

```

(WINCLR W6)(PRINTC " " W6)
(PRINTC "      Strike any key to continue . . ." W6)
(CLRBFI)(SETQ X (TYI W6))(SETQ STATUS 'LOR-DONE)
  (RETURN T)))
((THENP FACT
    (CAR POSSIBILITIES))
  (SETQ SUCCESS T)(SETQ STATUS 'KEEP-GOING)
  (PRINTC "      " WIN)(PR-TEXT FACT)
  (PRINTC "      was deduced from the following:" W4)
  (PRINTC " " W4)(SETQ NUMB 1)(SETQ NO-REA 0)
  (MAPCAR 'PR-SCREEN
    (CDR (CADDR (CAR POSSIBILITIES))))))
(SETQ POSSIBILITIES
  (CDR POSSIBILITIES))
  (GO LOOP)))

(DEF 'THENP
  '(LAMBDA (FACT RULE)
    (MEMBEROF FACT
      (CDAR (CDDR RULE)))))

(DEF 'INQUIRE
  '(LAMBDA ()
    (PROG ()
      (SETQ STATUS 'KEEP-GOING)
      AGAIN
      (WINCLR W6)(WINATTR 120 W6)
      (PRINTC " Enter line number of item you wish to question," W6)
      (PRINTC " else enter X" W6)
      (CLRBFI)(SETQ X (TYI W6))
      (COND
        ((OR (EQUAL X "X")(EQUAL X "x"))(SETQ STATUS 'LOR-DONE))
        ((GT (CHRVAL X)(+ 48 NO-REA))(BEEP)(GO AGAIN))
        ((EQUAL X "1")(HOW ONE))
        ((EQUAL X "2")(HOW TWO))
        ((EQUAL X "3")(HOW THR))
        ((EQUAL X "4")(HOW FOU))
        ((EQUAL X "5")(HOW FIV))
        ((EQUAL X "6")(HOW SIX))
        ((EQUAL X "7")(HOW SEV))
        ((EQUAL X "8")(HOW EIG))
        ((EQUAL X "9")(HOW NIN))
        (T (BEEP)(GO AGAIN)))
      (COND
        ((EQUAL STATUS 'KEEP-GOING)(GO AGAIN))
        ((EQUAL STATUS 'EXIT) NIL)
        ((EQUAL STATUS 'LOR-DONE) NIL))))
))

(DEF 'FIND
  '(LAMBDA (PATTERN)
    (PROG (FACTLIST)
      (SETQ FACTLIST FACTS)
      LOOP
      (COND
        ((NULL FACTLIST)(RETURN NIL))
        ((MATCH PATTERN (CAR FACTLIST))(RETURN (CAR FACTLIST)))
        (SETQ FACTLIST (CDR FACTLIST))
        (GO LOOP))))
```

```
(DEF 'MATCH
  '(LAMBDA (PAT ASSERT)
    (COND ((AND (NULL PAT)(OR (NULL ASSERT)(NUMBERP ASSERT))) T)
          ((OR (NULL PAT)(OR (NULL ASSERT)(NUMBERP ASSERT))) NIL)
          ((OR (EQUAL (CAR PAT) '+)(EQUAL (CAR PAT)(CAR ASSERT)))
           (MATCH (CDR PAT)(CDR ASSERT)))
          ((EQUAL (CAR PAT) '+)
           (COND ((MATCH (CDR PAT)(CDR ASSERT)))
                 ((MATCH PAT (CDR ASSERT)))))

(DEF 'ADD
  '(LAMBDA (NEW)
    (COND
      [(MEMB NEW FACTS)
       NIL]
      [T
       (SETQ FACTS
             (CONS NEW FACTS))
       (SETQ TEMPLIST
             (CONS NEW TEMPLIST))))])))

(DEF 'ANS
  '(LAMBDA (NO PHRASE WIND)
    [PROG (POS)
      (PRINC " " WIND)(PRINC " " WIND)
      (SETQ POS (WINCURS NIL WIND))(WINPOS POS W7)
      (PRINC NO W7)
      (WINCURS (LIST (CAR POS)(ADD1 (CADR POS))) WIND)
      (PRINC " " WIND)(PRINC PHRASE WIND)]))
```

FILE: CONV.LSP

```
(DEF 'CONVERT-CONSTRAINTS
  '(LAMBDA ()
    (PROG (RULE-CONVERSION)
      (SETQ RULE-CONVERSION
        '(((RULE 1 (IF (FINANCIAL RISK SHOULD BE LOW . 10))
          (THEN (FINANCIAL RISK IS LOW . 10)))
        (RULE 2 (IF (FINANCIAL RISK CAN BE MODERATE . 20))
          (THEN (FINANCIAL RISK IS MODERATE . 20)))
        (RULE 3 (IF (FINANCIAL RISK CAN BE HIGH . 30))
          (THEN (FINANCIAL RISK IS HIGH . 30)))
        (RULE 4 (IF (OBJECTIVE SHOULD BE INCOME . 5))
          (THEN (OBJECTIVE OF INCOME IS SATISFIED . 5)))
        (RULE 5 (IF (OBJECTIVE SHOULD BE RETIREMENT SECURITY . 10))
          (THEN (OBJECTIVE OF RETIREMENT SECURITY IS SATISFIED . 10)))
        (RULE 6 (IF (OBJECTIVE SHOULD BE GROWTH . 15))
          (THEN (OBJECTIVE OF GROWTH IS SATISFIED . 15)))
        (RULE 7 (IF (OBJECTIVE SHOULD BE SPECULATION . 20))
          (THEN (OBJECTIVE OF SPECULATION IS SATISFIED . 20)))
        (RULE 8 (IF (TAX EXEMPTION SHOULD BE HIGH . 4))
          (THEN (TAX ADVANTAGE IS HIGH . 4)))
        (RULE 9 (IF (TAX EXEMPTION IS MODERATELY IMPORTANT . 8))
          (THEN (TAX ADVANTAGE IS MODERATE . 8)))
        (RULE 10 (IF (TAX EXEMPTION IS NOT IMPORTANT . 12))
          (THEN (TAX ADVANTAGE IS LOW . 12)))
        (RULE 11 (IF (INFLATION PROTECTION SHOULD BE HIGH . 3))
          (THEN (INFLATION PROTECTION IS HIGH . 3)))
        (RULE 12 (IF (INFLATION PROTECTION IS MODERATELY IMPORTANT . 6))
          (THEN (INFLATION PROTECTION IS MODERATE . 6)))
        (RULE 13 (IF (INFLATION PROTECTION IS NOT A CONSIDERATION . 9))
          (THEN (INFLATION PROTECTION IS LOW . 9)))
        (RULE 14 (IF (LIQUIDITY SHOULD BE HIGH . 2))
          (THEN (LIQUIDITY IS HIGH . 2)))
        (RULE 15 (IF (LIQUIDITY SHOULD BE MODERATE . 4))
          (THEN (LIQUIDITY IS MODERATE . 4)))
        (RULE 16 (IF (LIQUIDITY IS NOT A CONSIDERATION . 6))
          (THEN (LIQUIDITY IS LOW . 6)))
        (RULE 17 (IF (EFFORT REQUIRED SHOULD BE LOW . 2))
          (THEN (EFFORT REQUIRED IS LOW . 2)))
        (RULE 18 (IF (EFFORT REQUIRED SHOULD BE MODERATE . 4))
          (THEN (EFFORT REQUIRED IS MODERATE . 4)))
        (RULE 19 (IF (EFFORT REQUIRED CAN BE HIGH . 6))
          (THEN (EFFORT REQUIRED IS HIGH . 6)))
        (RULE 20 (IF (SKILL REQUIRED SHOULD NOT BE DEMANDING . 2))
          (THEN (SKILL REQUIRED IS NOT DEMANDING . 2)))
        (RULE 21 (IF (SKILL REQUIRED CAN BE MODERATE . 4))
          (THEN (SKILL REQUIRED IS MODERATE . 4)))
        (RULE 22 (IF (SKILL REQUIRED CAN BE HIGH . 6))
          (THEN (SKILL REQUIRED IS HIGH . 6)))))

      (FORWARDCHAIN RULE-CONVERSION)
      (SETQ RULE-CONVERSION NIL))))
```

FILE: C-RULE.LSP

```
<DEF 'CONSTRAINT-RULELIST
  '(LAMBDA ()
    (PROG (CLAUS CONSTRAINTS LIQ-DEFAULT SKILL-DEFAULT)
      (SETQ CONSTRAINTS
        '((RULE 101
            (IF (YOU INDICATED AN OBJECTIVE OF GROWTH))
            (THEN (OBJECTIVE SHOULD BE GROWTH . 15)))
          (RULE 102
            (IF (YOU INDICATED AN OBJECTIVE OF RETIREMENT SECURITY))
            (THEN (OBJECTIVE SHOULD BE RETIREMENT SECURITY . 10)))
          (RULE 103
            (IF (YOU INDICATED AN OBJECTIVE OF INCOME))
            (THEN (OBJECTIVE SHOULD BE INCOME . 5)))
          (RULE 104
            (IF (YOU INDICATED AN OBJECTIVE OF SPECULATION))
            (THEN (OBJECTIVE SHOULD BE SPECULATION . 20)))
          (RULE 154
            (IF (INCOME IS VERY SECURE)
                (EMERGENCY RESERVES ARE ADEQUATE))
            (THEN (LIQUIDITY IS NOT A CONSIDERATION . 6)))
          (RULE 155
            (IF (INCOME IS MODERATELY SECURE)
                (EMERGENCY RESERVES ARE INADEQUATE)
                (YOU ARE A MODERATE RISK-TAKER))
            (THEN (LIQUIDITY IS NOT A CONSIDERATION . 6)))
          (RULE 156
            (IF (INCOME IS MODERATELY SECURE)
                (EMERGENCY RESERVES ARE INADEQUATE)
                (YOU ARE AN AGGRESSIVE RISK-TAKER))
            (THEN (LIQUIDITY SHOULD BE MODERATE . 4
                  )))
          (RULE 157
            (IF (INCOME IS NOT VERY SECURE)
                (EMERGENCY RESERVES ARE ADEQUATE)
                (YOU ARE A CONSERVATIVE RISK-TAKER))
            (THEN (LIQUIDITY SHOULD BE HIGH . 3)))
          (RULE 158
            (IF (INCOME IS NOT VERY SECURE)
                (EMERGENCY RESERVES ARE ADEQUATE)
                (YOU ARE A MODERATE RISK-TAKER))
            (THEN (LIQUIDITY SHOULD BE MODERATE . 4
                  )))
          (RULE 201
            (IF (TAXABLE INCOME IS LESS THAN $25000))
            (THEN (TAX EXEMPTION IS NOT IMPORTANT . 12)))
          (RULE 202
            (IF (TAXABLE INCOME IS BETWEEN $25000 AND $45000)
                )
            (THEN (TAX EXEMPTION IS MODERATELY IMPORTANT . 8)))
          (RULE 203
            (IF (TAXABLE INCOME IS MORE THAN $45000))
            (THEN (TAX EXEMPTION SHOULD BE HIGH . 4)))
          (RULE 251
            (IF (ANTICIPATED INFLATION IS LOW))
            (THEN (INFLATION PROTECTION IS NOT A CONSIDERATION . 9)))
          (RULE 252
            (IF (ANTICIPATED INFLATION IS MODERATE))
            (THEN (INFLATION PROTECTION IS MODERATELY IMPORTANT . 8))))
```

```

(RULE 253
  (IF (ANTICIPATED INFLATION IS HIGH))
  (THEN (INFLATION PROTECTION SHOULD BE HIGH . 3)))
(RULE 301
  (IF (DESIRE TO PARTICIPATE IS LOW))
  (THEN (EFFORT REQUIRED SHOULD BE LOW . 2)))
(RULE 302
  (IF (DESIRE TO PARTICIPATE IS MODERATE))
  (THEN (EFFORT REQUIRED CAN BE MODERATE . 4)))
(RULE 303
  (IF (DESIRE TO PARTICIPATE IS HIGH))
  (THEN (EFFORT REQUIRED CAN BE HIGH . 6)))
(RULE 354
  (IF (EXPERIENCE LEVEL IS HIGH)
    (FINANCIAL SERVICE IS CURRENTLY USED))
  (THEN (SKILL REQUIRED CAN BE HIGH . 6)))
(RULE 355
  (IF (EXPERIENCE LEVEL IS MODERATE)
    (FINANCIAL SERVICE IS CURRENTLY USED))
  (THEN (SKILL REQUIRED CAN BE MODERATE . 4)))
(RULE 401
  (IF (YOU ARE A CONSERVATIVE RISK-TAKER))
  (THEN (FINANCIAL RISK SHOULD BE LOW . 10)))
(RULE 402
  (IF (YOU ARE A MODERATE RISK-TAKER))
  (THEN (FINANCIAL RISK CAN BE MODERATE . 20)))
(RULE 403
  (IF (YOU ARE AN AGGRESSIVE RISK-TAKER))
  (THEN (FINANCIAL RISK CAN BE HIGH . 30)))))

(SETQ LIQ-DEFAULT
  '((RULE 151
    (IF (EMERGENCY RESERVES ARE INADEQUATE))
    (THEN (LIQUIDITY SHOULD BE HIGH . 2)))
  (RULE 152
    (IF (EMERGENCY RESERVES ARE MODERATELY ADEQUATE))
    (THEN (LIQUIDITY SHOULD BE MODERATE . 4)))
  (RULE 153
    (IF (EMERGENCY RESERVES ARE ADEQUATE))
    (THEN (LIQUIDITY IS NOT A CONSIDERATION . 6))))))

(SETQ SKILL-DEFAULT
  '((RULE 351
    (IF (EXPERIENCE LEVEL IS HIGH))
    (THEN (SKILL REQUIRED CAN BE MODERATE . 4)))
  (RULE 352
    (IF (EXPERIENCE LEVEL IS MODERATE))
    (THEN (SKILL REQUIRED SHOULD NOT BE DEMANDING . 2)))
  (RULE 353
    (IF (EXPERIENCE LEVEL IS MINIMAL))
    (THEN (SKILL REQUIRED SHOULD NOT BE DEMANDING . 2))))))

(FORWARDCHAIN CONSTRAINTS)
(SETQ CONSTRAINTS NIL)
(COND ((NOT (FIND '(LIQUIDITY +)))
       (FORWARDCHAIN LIQ-DEFAULT)))
(COND ((NOT (FIND '(SKILL REQUIRED +)))
       (FORWARDCHAIN SKILL-DEFAULT)))
(SETQ SKILL-DEFAULT NIL)
(SETQ LIQ-DEFAULT NIL)
(SETQ CLAUS (FIND '(OBJECTIVE +)))
(STORE (CDR 0) (CDR (LAST CLAUS))))

```

```
(SETQ CLAUS (FIND '(FINANCIAL RISK +)))
  (STORE (GOAL 1) (CDR (LAST CLAUS)))
(SETQ CLAUS (FIND '(LIQUIDITY +)))
  (STORE (GOAL 2) (CDR (LAST CLAUS)))
(SETQ CLAUS (FIND '(TAX EXEMPTION +)))
  (STORE (GOAL 3) (CDR (LAST CLAUS)))
(SETQ CLAUS (FIND '(INFLATION +)))
  (STORE (GOAL 4) (CDR (LAST CLAUS)))
(SETQ CLAUS (FIND '(EFFORT +)))
  (STORE (GOAL 5) (CDR (LAST CLAUS)))
(SETQ CLAUS (FIND '(SKILL +)))
  (STORE (GOAL 6) (CDR (LAST CLAUS))))])
```

FILE: DO-INS.LSP

```

(DEF 'DO-INSURANCE
  '(LAMBDA ()
    (PROG (FUNERAL C-DEBT EMERG C-FUND FEE NO-COL LIVING NINETY RATE
           SPOUSE SOC-SEC NET TOT-LIV TOT-MON ASSETS NEEDS PORT)
      START
        (SETQ MODE 'INS-MODE)
        (WAIT)
        (LOAD "B:INS.LSP")
        (STOP-WAIT)
        (INS-INTERVIEW)
        (WAIT)
        (DEF 'INS-INTERVIEW NIL)
        (COND ((EQUAL MODE 'INV-MODE) (RETURN NIL)))
        (LOAD "B:I-RULE.LSP")
        (INS-RULELIST)
        (DEF 'INS-RULELIST NIL)
        (STOP-WAIT)
        (SHOW-INS W1) (CHOICE)
        (WAIT)
        (SETQ FACTS NIL) (SETQ RULES-USED NIL)
        (DEF 'INQUIRE-INS NIL) (DEF 'SHOW-INS NIL)))))

(DEF 'SHOW-INS
  '(LAMBDA (WIND)
    (WINCLR WIND)
    (PRINTC " ESTIMATING YOUR LIFE INSURANCE NEEDS:" WIND)
    (PRINTC " * " WIND)
    (PRINTC " i - Funeral, estate taxes, etc.          $" WIND)
    (PRINC FUNERAL WIND)
    (PRINTC " 2 - Settle non-mortgage debt (consumer debt)      " WIND)
    (PRINC C-DEBT WIND)
    (PRINTC " 3 - Emergency fund                      " WIND)
    (PRINC EMERG WIND)
    (PRINTC " 4 - College fund                      " WIND)
    (PRINC C-FUND WIND)
    (PRINTC " 5 - Expected living expenses:" WIND)
    (PRINTC "   a - Average annual living expenses      $" WIND)
    (PRINC LIVING WIND)
    (PRINTC "   b - Spouse's average annual income      " WIND)
    (PRINC SPOUSE WIND)
    (PRINTC "   c - Annual Social Security benefits      " WIND)
    (PRINC SOC-SEC WIND)
    (PRINTC "   d - Net annual living expenses (a - b - c)" WIND)
    (PRINC NET WIND)
    (PRINTC "   e - Investment rate factor            " WIND)
    (PRINC RATE WIND)
    (PRINTC "   f - Total living expenses needed (d x e)" WIND)
    (PRINC TOT-LIV WIND)
    (PRINTC " ")
    --- WIND)
    (PRINTC " g - Total monetary needs (1+2+3+4+f)" $" WIND)
    (PRINC TOT-MON WIND)
    (PRINTC " h - Total investment assets in hand      " WIND)
    (PRINC ASSETS WIND)
    (PRINTC " ")
    --- WIND)
    (PRINTC " i - Life insurance needs (g - h)" $" WIND)
    (PRINC NEEDS WIND)))

```

```

<DEF 'INQUIRE-INS
  '(LAMBDA ()
    '(PROG ()
      AGAIN
      (WINCLR W6)(WINATTR 120 W6)
      (PRINTC " Enter line number of item you wish to question," W6)
      (PRINTC " else enter X" W6)
      (CLRBFI)(SETQ X (TYI W6))
      (COND
        ((OR (EQUAL X "X")(EQUAL X "x"))(SETQ STATUS 'EXIT))
        ((EQUAL X "1")(HOW (FIND '(YOUR FUNERAL EXPENSES ARE ESTIMATED TO BE
      >>>
        ((EQUAL X "2")(HOW '(YOU MUST INCLUDE NON-MORTGAGE DEBT IN TOTAL MONETARY NEEDS)))
        ((EQUAL X "3")(HOW '(YOU MUST INCLUDE EMERGENCY FUND REQUIREMENT IN TOTAL MONETARY NEEDS)))
        ((EQUAL X "4")(HOW (FIND '(THERE IS + COLLEGE FUND))))
        ((EQUAL X "5")(HOW '(YOU MUST INCLUDE TOTAL LIVING EXPENSES IN TOTAL MONETARY NEEDS)))
        ((OR (EQUAL X "A")(EQUAL X "a"))
          (HOW '(AVERAGE ANNUAL LIVING EXPENSES ARE CONSIDERED)))
        ((OR (EQUAL X "B")(EQUAL X "b"))
          (HOW '(SPOUSES AVERAGE ANNUAL INCOME IS SUBTRACTED FROM ANNUAL
      LIVING EXPENSES)))
        ((OR (EQUAL X "C")(EQUAL X "c"))
          (HOW (FIND '(SOCIAL SECURITY BENEFITS WILL BE > PER YEAR))))
        ((OR (EQUAL X "D")(EQUAL X "d"))
          (HOW '(NET ANNUAL LIVING EXPENSES ARE CONSIDERED)))
        ((OR (EQUAL X "E")(EQUAL X "e"))
          (HOW (FIND '(INVESTMENT RATE FACTOR IS >))))
        ((OR (EQUAL X "F")(EQUAL X "f"))
          (HOW '(YOU MUST INCLUDE TOTAL LIVING EXPENSES IN TOTAL MONETARY NEEDS)))
        ((EQUAL X "6")(HOW '(YOU MUST CONSIDER TOTAL MONETARY NEEDS)))
        ((EQUAL X "7")(HOW '(SUBTRACT INVESTMENT ASSETS IN HAND FROM TOTAL MONETARY NEEDS)))
        ((EQUAL X "8")(HOW '(LIFE INSURANCE NEEDS SHOULD BE ESTIMATED)))
        (T (BEEP)(GO AGAIN)))))))

```

FILE: DO-INV.LSP

```

(DEF 'DO-INVESTMENTS
  '(LAMBDA ()
    (PROG ()
      (SETQ MODE 'CONSTR-MODE)
      (LOAD "B:QUERY.LSP")
      (STOP-WAIT)
      (INV-INTERVIEW)
      (WAIT)
      (DEF 'INV-INTERVIEW NIL)
      (SETQ TEMPLIST NIL)
      (LOAD "B:C-RULE.LSP")
      (CONSTRAINT-RULELIST)
      (DEF 'CONSTRAINT-RULELIST NIL)
      (SETQ CLIENT-CONSTRAINTS TEMPLIST)
      (STOP-WAIT)
      (SETQ WIN W1)(SHOW-CONSTR)
      (CHOICE)
      (WAIT)
      (SETQ TEMPLIST NIL)
      (SETQ CLIENT-CONSTRAINTS NIL)
      (LOAD "B:CONV.LSP")
      (CONVERT-CONSTRAINTS)
      (DEF 'CONVERT-CONSTRAINTS NIL)
      (SETQ MODE 'INV-MODE)
      (LOAD "B:I-PAR.LSP")
      (INV-PARAM)
      (DEF 'INV-PARAM NIL)
      (SETQ TEMPLIST NIL)
      (LOAD "B:INVEST.LSP")
      (LOAD "B:RANK.LSP")
      (RANK-INV)
      (SORT-INV)
      (SETQ INVESTMENT-RULES NIL)
      (DEF 'INVESTMENT-RULELIST NIL)
      (DEF 'RANK-INV NIL)(DEF 'EVAL-INV NIL)
      (DEF 'SORT NIL)(DEF 'SORT-INV NIL)
      (STOP-WAIT)
      (SETQ WIN W1)(SHOW-INV)
      (CHOICE)]))

(DEF 'SHOW-CONSTR
  '(LAMBDA ()
    (WINCLR WIN)
FOR YOU:  WINPRINTC "THE FOLLOWING INVESTMENT CONSTRAINTS HAVE BEEN ESTABLISHED"
    (PRINTC " " WIN)(SETQ NUMB 1)(SETQ NO-REA 0)
    (MAPC 'PR-SCREEN CLIENT-CONSTRAINTS)))

(DEF 'INQUIRE-CONSTR
  '(LAMBDA ()
    (PROG ()
      AGAIN
      (WINCLR W6)(WINATTR 120 W6)
      (PRINTC " Enter line number of item you wish to question." W6)
      (PRINTC " else enter X" W6)
      (CLRSFI)(SETQ X (TYI W6))
      (COND
        
```

```

((OR (EQUAL X "X") (EQUAL X "x")) (SETQ STATUS 'EXIT))
((GT (CHRVAL X) (+ 48 NO-REA)) (BEEP) (GO AGAIN))
((EQUAL X "1") (HOW ONE))
((EQUAL X "2") (HOW TWO))
((EQUAL X "3") (HOW THREE))
((EQUAL X "4") (HOW FOUR))
((EQUAL X "5") (HOW FIVE))
((EQUAL X "6") (HOW SIX))
((EQUAL X "7") (HOW SEVEN))
((EQUAL X "8") (HOW EIGHT))
((EQUAL X "9") (HOW NINE))
(T (BEEP) (GO AGAIN)))]])

(DEF 'SHOW-INV
  '(LAMBDA ()
    (WINCLR WIN)
    (PRINTC " You have $" WIN) (PRINC CAPITAL WIN)
    (PRINC " available for investment. Based upon" WIN)
    (PRINTC " your situation, temperament, and goals, FINPLAN recommends")
    (COND ((EQUAL NO-INV 1) (PRINC " that your portfolio includes investments allocated to the"))
    (e" WIN) . (PRINTC " investment classes listed below." WIN)
    (PRINTC " " WIN)
    (PRINTC " 1 -" WIN) (PRINC NO-INV WIN)
    (COND ((EQUAL NO-INV 1) (PRINC " investment class is recommended." WIN))
    (T (PRINC " investment classes are recommended." WIN)))
    (PRINTC " " WIN) (PRINTC " 2 - $" WIN) (PRINC UNIT WIN)
    (PRINC " is recommended for investment in each class." WIN)
    (PRINTC " " WIN) (PRINTC " 3 - Class 1: $" WIN)
    (PRINC UNIT WIN) (PRINC " in " WIN) (PR-TEXT FIRST)
    (COND ((GE NO-INV 2)
      (PRINTC " " WIN) (PRINTC " 4 - Class 2: $" WIN)
      (PRINC UNIT WIN) (PRINC " in " WIN) (PR-TEXT SECOND)))
    (COND ((GE NO-INV 3)
      (PRINTC " " WIN) (PRINTC " 5 - Class 3: $" WIN)
      (PRINC UNIT WIN) (PRINC " in " WIN) (PR-TEXT THIRD)))
    (COND ((EQUAL NO-INV 4)
      (PRINTC " " WIN) (PRINTC " 6 - Class 4: $" WIN)
      (PRINC UNIT WIN) (PRINC " in " WIN) (PR-TEXT FOURTH)))
    (COND ((GT REMAIN 1)
      (PRINTC " " WIN) (PRINTC " Place the remaining $" WIN)
      (PRINC REMAIN WIN) (PRINC " in your savings." WIN))))))

(DEF 'INQUIRE-INV
  '(LAMBDA ()
    [PROG ()
      AGAIN
      (WINCLR W6) (WINATTR 120 W6)
      (PRINTC " Enter line number of item you wish to question." W6)
      (PRINTC " else enter X" W6)
      (CLREFI) (GETO K (TYI W6))
      (COND
        ((OR (EQUAL X "X") (EQUAL X "x")) (SETQ STATUS 'EXIT))
        ((GT (CHRVAL X) (+ 50 NO-INV)) (BEEP) (GO AGAIN))
        ((EQUAL X "1") (HOW (FIND "/> INVESTMENT +/+/))
        ((EQUAL X "2") (HOW (FIND "/> OF ORIGINAL CAPITAL +/+/)
        ((EQUAL X "3") (HOW FIRST))
        ((EQUAL X "4") (HOW SECOND))
        ((EQUAL X "5") (HOW THIRD))
        ((EQUAL X "6") (HOW FOURTH))
        T (BEEP) (GO AGAIN))]])

```

FILE: INIT.LSP

```
<DEF 'INITIALIZE
  '(LAMBDA ()
    (WINCLR)
    (SETQ W1 (WINDOW '(0 0) '(23 80)))
    (SETQ W2 (WINDOW '(24 30) '(1 20)))
    (SETQ W3 (WINDOW '(24 20) '(1 40)))
    (SETQ W4 (WINDOW '(2 2) '(19 76)))
    (SETQ W5 (WINDOW '(0 30) '(1 20)))
    (SETQ W6 (WINDOW '(21 10) '(2 60)))
    (SETQ W7 (WINDOW '(0 0) '(1 1)))
    (WINATTR 120 W7)
    (WAIT)
    (SETQ MODE NIL)
    (SETQ FACTS NIL)
    (SETQ TEMPLIST NIL)
    (SETQ RULES-USED NIL)
    (SETQ GOAL (ARRAY 1 7))))
```

FILE: INS.LSP

```

(DEF 'INS-INTERVIEW
  '(LAMBDA ()
    (PROG ()
      (ADD '(YOU WANT TO ESTIMATE LIFE INSURANCE NEEDS))
      (WINCLR W1)
      (PRINTC
        "In order to estimate your life insurance needs, please answer the following questions." W1)
      (PRINTC "the following questions." W1)
      (PRINTC " " W1)
      DEPEND
      (PRINTC "How many dependents (or others) do you wish to support in the event of your death?" W1)
      (PRINTC " " W1)(PRINTC " " W1)
      (SETQ X (READ W1))
      (COND [(NOT (NUMBERP X))(BEEP)(GO DEPEND)]
             [(EQUAL X 0)(SETQ MODE 'INU-MODE)(PRINTC " " W1)
              (PRINTC "You have no need for life insurance." W1)
              (WINATTR 120 W3)
              (PRINTC "Strike any key to continue . . ." W3)
              (CLRBFI)(TYI W3)(WINCLR)(RETURN NIL)]
             [(PLUSP X) NIL]
             [T (BEEP)(GO DEPEND)])
      (WINCLR W1)
      GROSS
      (PRINTC "What is the amount of your gross estate?" W1)
      (PRINTC "(net worth and anticipated insurance proceeds)??" W1)
      (ANS '1 "Under $20,000" W1)
      (ANS '2 "Between $20,000 and $200,000" W1)
      (ANS '3 "More than $200,000" W1)(PRINTC " " W1)(PRINTC " " W1)
      (SETQ X
            (READ W1))
      (COND
        [(NOT (NUMBERP X))
         (BEEP)
         (GO GROSS)]
        [(EQUAL X 1)
         (SETQ FUNERAL 2200)
         (ADD '(YOUR GROSS ESTATE IS UNDER $20000))]
        [(EQUAL X 2)
         (SETQ FUNERAL 5000)
         (ADD '(YOUR GROSS ESTATE IS BETWEEN $20000 AND $200000))]
        [(EQUAL X 3)
         (SETQ FUNERAL 10000)
         (ADD '(YOUR GROSS ESTATE IS MORE THAN $200000))]
        [T
         (BEEP)
         (GO GROSS)])
      (WINCLR W1)
      CONSUMER
      (PRINTC
        "What is your approximate outstanding consumer debt (all non-mortgage debts)??" W1)
      (PRINTC "debt such as car loan, credit card, or other similar debt" W1)
    )))
  )
)
```

```

(PRINTC " " W1)(PRINTC " " W1)
(SETQ X
  (READ W1))
(COND
  [(NOT (NUMBERP X))
   (BEEP)
   (GO CONSUMER)]
  [(GT X 50000)
   (PRINTC "Too much - try again" W1)
   (GO CONSUMER)]
  [T
   (SETQ C-DEBT X)])
(WINCLR W1)
EMERGENCY
(PRINTC
  "Enter a sufficient sum for what you ought to keep as an emergency"
  " W1)
(PRINTC "reserve. (If you don't know, estimate two month's take-h"
  "ome pay.)* W1)
(PRINTC " " W1)(PRINTC " " W1)
(SETQ X
  (READ W1))
(COND
  [(NOT (NUMBERP X))
   (BEEP)
   (GO EMERGENCY)]
  [(GT X 50000)
   (PRINTC "I don't believe it - try again" W1)
   (GO EMERGENCY)]
  [T
   (SETQ EMERG X)])
(WINCLR W1)
CHILDREN
(PRINTC "Do you have any children?" W1)
(PRINTC " " W1)(PRINTC " " W1)
(SETQ X (READ W1))
(COND
  [(NUMBERP X)(BEEP)(GO CHILDREN)]
  [(OR (EQUAL (CHRVAL X) 89)(EQUAL (CHRVAL X) 121))
   (PROG ()
     COLLEGE
     (PRINTC
       "Do you intend to provide a college fund for your children?"
       " W1)
     (PRINTC " " W1)(PRINTC " " W1)
     (SETQ X
       (READ W1))
     (COND
       [(NUMBERP X)(BEEP)(GO COLLEGE)]
       [(OR (EQUAL (CHRVAL X) 89)(EQUAL (CHRVAL X) 121))
        (PROG ()
          (ADD '(YOU INTEND TO PROVIDE A COLLEGE FUND))
          SCHOOL
          (PRINTC "What kind of school do you have in mind?" W1)
          (ANS '1 "Private" W1)
          (ANS '2 "Public" W1)(PRINTC " " W1)(PRINTC " " W1)
          (SETQ X
            (READ W1))
          (COND
            [(NOT (NUMBERP X))
             (BEEP)
             (GO SCHOOL)])]
       )]
    )]
  )
)
```

```

[ (EQUAL X 1)(SETQ FEE 36000)
  (ADD '(YOU HAVE A PRIVATE COLLEGE IN MIND)) ]
[ (EQUAL X 2)(SETQ FEE 20000)
  (ADD '(YOU HAVE A PUBLIC COLLEGE IN MIND)) ]
[ T
  (BEEP)
  (GO SCHOOL) ]

CHILD-COL
(PRINTC "How many children?" W1)(PRINTC " " W1)
(PRINTC " " W1)(SETQ NO-COL (READ W1))
(COND
  [ (NOT (NUMBERP NO-COL))(BEEP)(GO CHILD-COL)]
  [ (GT NO-COL 11)(BEEP)(PRINTC " I don't believe it." W1)
    (GO CHILD-COL) ]
  [ (SETQ C-FUND (* FEE NO-COL)) ]
[ (OR (EQUAL (CHRVAL X) 78)(EQUAL (CHRVAL X) 110))(SETQ C-FUND 0)
  (ADD '(YOU DO NOT INTEND OR NEED TO PROVIDE A COLLEGE FUND)) ]
[ T
  (BEEP)
  (GO COLLEGE) ] )
[ (OR (EQUAL (CHRVAL X) 79)(EQUAL (CHRVAL X) 110))
  (SETQ C-FUND 0))(ADD '(YOU DO NOT INTEND OR NEED TO P~ RVIDE A COLLEGE FUND)) ]
[ T (BEEP)(GO CHILDREN) ]
(WINCLR W1)

LIVING-EXP
(PRINTC
  "Estimate the average annual living expenses required by yo~ ur survivors." W1)
(PRINTC
  "(Start with your current spending habits, allow a reduction~ n in some" W1)
(PRINTC
  " living expenses, anticipate possible new child-care costs.)" W1)
(PRINTC " " W1)(PRINTC " $" W1)
(SETQ X
  (READ W1))
(COND
  [ (NOT (NUMBERP X))
    (BEEP)
    (GO LIVING-EXP) ]
  [ (LT X 15000)
    (PRINTC "Too low - try again" W1)
    (GO LIVING-EXP) ] )
[ T
  (SETQ LIVING X) ]
(WINCLR W1)

SPS
(PRINTC "Do you have a spouse?" W1)
(PRINTC " " W1)(PRINTC " " W1)
(SETQ X (READ W1))
(COND
  [ (NUMBERP X)(BEEP)(GO SPS) ]
  [ (OR (EQUAL (CHRVAL X) 89)(EQUAL (CHRVAL X) 121)) NIL]
  [ (OR (EQUAL (CHRVAL X) 78)(EQUAL (CHRVAL X) 110))
    (SETQ SPOUSE 0)(WINCLR W1)(GO MINOR) ]
[ T (BEEP)(GO SPS) ] )

SPOUSE-WORK
(PRINTC "Does your spouse currently work for pay?" W1)
(PRINTC " " W1)(PRINTC " " W1)
(SETQ X
  (READ W1))

```

```

(COND
  [((NUMBERP X))(BEEP)(GO SPOUSE-WORK)]
  [(OR (EQUAL (CHRVAL X) 89)(EQUAL (CHRVAL X) 121))(GO SPOUSE-PAY)]
  [(OR (EQUAL (CHRVAL X) 78)(EQUAL (CHRVAL X) 110))(GO SPOUSE-INC)]
  [T (BEEP)(GO SPOUSE-WORK)])
    SPOUSE-PAY
    (PRINTC "Estimate your spouse's average after-tax salary" W1)
    (PRINTC " " W1)(PRINTC " $" W1)
    (SETQ X
      (READ W1))
  (COND
    [((NOT (NUMBERP X)))
     (BEEP)
     (GO SPOUSE-PAY)]
    [(> X 100000)(BEEP)(PRINTC "I don't believe it." W1)
     (GO SPOUSE-PAY)]
    [T
      (SETQ SPOUSE X)(WINCLR W1)(GO MINOR)])
  SPOUSE-INC
  (PRINTC
    "Estimate your spouse's average after-tax salary per "
    year if she went" W1)
  (PRINTC "back to work today." W1)
  (PRINTC " " W1)(PRINTC " $" W1)
  (SETQ X
    (READ W1))
  (COND
    [((NOT (NUMBERP X)))
     (BEEP)
     (GO SPOUSE-INC)]
    [(> X 100000)(BEEP)(PRINTC "I don't believe it." W1)
     (GO SPOUSE-INC)]
    [T
      (SETQ SPOUSE X)])
  (WINCLR W1)
MINOR
  (PRINTC "How many minor children do you have?" W1)
  (PRINTC " " W1)(PRINTC " " W1)
  (SETQ X
    (READ W1))
  (COND
    [((NOT (NUMBERP X)))
     (BEEP)
     (GO MINOR)]
    [(> X 1)
     (SETQ SOC-SEC 5000)
     (ADD '(YOU HAVE TWO OR MORE MINOR CHILDREN))]
    [(> X 0)
     (SETQ SOC-SEC 4000)
     (ADD '(YOU HAVE ONE MINOR CHILD))]
    [(EQUAL X 0)
     (SETQ SOC-SEC 3000)
     (ADD '(YOU HAVE NO MINOR CHILDREN))]
    [T
      (BEEP)
      (GO MINOR)])
  (WINCLR W1)
PORTFOLIO
  (PPINTC
    "How do you expect your spouse (or other primary beneficiary)" W1)
  (PRINTC "to invest the insurance proceeds upon your death?" W1)
  (ANS '1 'In a conservative manner /bank accounts and bonds'

```

```

(ANS '2 "In a more aggressive manner (stocks and real estate)" * W1)
  (PRINTC " " W1)(PRINTC " " W1)
  (SETQ X
    (READ W1))
  (COND
    [(NOT (NUMBERP X))
      (BEEP)
      (GO PORTFOLIO)]
    [(EQUAL X 1)
      (SETQ PORT
        'CONSERVATIVE)
      (ADD '(INSURANCE PROCEEDS WILL BE INVESTED IN A
            CONSERVATIVE PORTFOLIO))]
    [(EQUAL X 2)
      (SETQ PORT
        'AGGRESSIVE)
      (ADD '(INSURANCE PROCEEDS WILL BE INVESTED IN AN
            AGGRESSIVE PORTFOLIO))]
    [T
      (BEEP)
      (GO PORTFOLIO)])

    (WINCLR W1)
    GET-SPOUSE-AGE
    (PRINTC "What is the current age of your spouse (or other primary"
beneficiary)? " W1)
    (PRINTC " " W1)(PRINTC " " W1)
    (SETQ X
      (READ W1))
    (COND
      [(NOT (NUMBERP X))
        (BEEP)
        (GO GET-SPOUSE-AGE)]
      [(GT X 100)(BEEP)(PRINTC "TRY AGAIN" W1)(GO GET-SPOUSE-AGE)]
      [T
        (SETQ SPOUSE-AGE X)
        (SETQ NINETY
          (- 90 X))])
    (COND
      [(LT NINETY 28)
        (SETQ RATE 20)
        (ADD '(YOUR SPOUSE WILL BE 90 IN ABOUT 25 YEARS))]
      [(LT NINETY 33)
        (SETQ RATE 22)
        (ADD '(YOUR SPOUSE WILL BE 90 IN ABOUT 30 YEARS))]
      [(LT NINETY 38)
        (SETQ RATE 25)
        (ADD '(YOUR SPOUSE WILL BE 90 IN ABOUT 35 YEARS))]
      [(LT NINETY 43)
        (SETQ RATE 27)
        (ADD '(YOUR SPOUSE WILL BE 90 IN ABOUT 40 YEARS))]
      [(LT NINETY 48)
        (SETQ RATE 30)
        (ADD '(YOUR SPOUSE WILL BE 90 IN ABOUT 45 YEARS))]
      [(LT NINETY 53)
        (SETQ RATE 31)
        (ADD '(YOUR SPOUSE WILL BE 90 IN ABOUT 50 YEARS))]
      [(LT NINETY 58)
        (SETQ RATE 33)
        (ADD '(YOUR SPOUSE WILL BE 90 IN ABOUT 55 YEARS))]
      [T
        (SETQ RATE 35)

```

```

(ADD '(YOUR SPOUSE WILL BE 90 IN ABOUT 60 YEARS OR MORE)
  )
(COND
  ((EQUAL PORT
    'AGGRESSIVE)
   (COND
     ((LT NINETY 28)
      (SETQ RATE 16))
     ((LT NINETY 33)
      (SETQ RATE 17))
     ((LT NINETY 38)
      (SETQ RATE 19))
     ((LT NINETY 43)
      (SETQ RATE 20))
     ((LT NINETY 48)
      (SETQ RATE 21))
     ((LT NINETY 53)
      (SETQ RATE 21))
     ((LT NINETY 58)
      (SETQ RATE 22))
     (T
      (SETQ RATE 23)))
   ((EQUAL PORT
     'CONSERVATIVE)
    )
    (T
      (PRINTC "ERROR - NO RATE")))
  (WINCLR W1)
  ASSETS-L
  (PRINTC "What is the total value of investment assets in hand?" W1)
  (PRINTC "(Include all assets that are invested or may become avail~
able for" W1)
  (PRINTC "investment, ie. checking, savings, earned pension benefit~
s," W1)
  (PRINTC "college fund you've already started, etc.)" W1)
  (PRINTC " " W1)(PRINTC " " W1)
  (SETQ ASSETS
    (READ W1))
  (COND
    ((NOT (NUMBERP ASSETS))(BEEP)(GO ASSETS-L))
    ((GT ASSETS 1000000)(BEEP)
      (PRINTC "You don't need this program. Go hire an accountant.
      (GO ASSETS-L)))
  (SETQ LUMP
    (+ FUNERAL C-DEBT))
  (SETQ LUMP
    (+ LUMP EMERG))
  (SETQ LUMP
    (+ LUMP C-FUND))
  (SETQ NET
    (- LIVING SPOUSE))
  (SETQ NET
    (- NET SOC-SEC))
  (SETQ TOT-LIV
    (- NET RATE))
  (SETQ TOT-MON
    (+ LUMP TOT-LIV))
  (SETQ NEEDS
    (- TOT-MON ASSETS))))])

```

FILE: INTRO.LSP

```

(DEF 'INTRO
  '(LAMBDA ()
    (PROG (A B C D E F)
      (STOP-WAIT)
      (SETQ A (WINDOW '(6 9) '(7 2)))
      (SETQ B (WINDOW '(6 11) '(1 4)))
      (SETQ C (WINDOW '(7 22) '(1 2)))
      (SETQ D (WINDOW '(7 29) '(4 2)))
      (SETQ E (WINDOW '(9 39) '(3 2)))
      (SETQ F (WINDOW '(1 5) '(16 70)))
      (WINATTR 120 F)(WINCLR F)(WINCURS '(2 30) F)
      (PRINC "WELCOME TO" F)(WINCLR A)(WINCLR B)
      (WINPOS '(9 11) B)(WINCLR B)
      (WINPOS '(6 17) A)(WINCLR A)
      (WINPOS '(6 21) A)(WINCLR A)(WINCLR C)
      (WINPOS '(8 23) C)(WINCLR C)
      (WINPOS '(9 24) C)(WINCLR C)
      (WINPOS '(10 25) C)(WINCLR C)
      (WINPOS '(11 26) C)(WINCLR C)
      (WINPOS '(6 27) A)(WINCLR A)
      (WINPOS '(6 31) A)(WINCLR A)
      (WINPOS '(6 33) B)(WINCLR B)
      (WINPOS '(9 33) B)(WINCLR B)
      (WINPOS '(6 36) D)(WINCLR D)
      (WINPOS '(6 40) A)(WINCLR A)
      (WINPOS '(12 42) B)(WINCLR B)
      (WINPOS '(6 51) B)(WINCLR B)
      (WINPOS '(7 50) E)(WINCLR E)
      (WINPOS '(7 54) E)(WINCLR E)
      (WINPOS '(9 49) B)(WINCLR B)(WINPOS '(9 53) C)(WINCLR C)
      (WINPOS '(9 53) C)(WINCLR C)
      (WINPOS '(10 48) E)(WINCLR E)
      (WINPOS '(10 56) E)(WINCLR E)
      (WINPOS '(6 60) A)(WINCLR A)
      (WINPOS '(7 61) C)(WINCLR C)
      (WINPOS '(9 62) C)(WINCLR C)
      (WINPOS '(9 63) C)(WINCLR C)
      (WINPOS '(10 64) C)(WINCLR C)
      (WINPOS '(11 65) C)(WINCLR C)
      (WINPOS '(6 66) A)(WINCLR A)
      (WINCURS '(14 11) F)
      (PRINC "AN EXPERT SYSTEM FOR PERSONAL FINANCIAL PLANNING" F)
      (WINCURS '(1? 16) NIL)(PRINC "SELECT: ")
      (SEL '1 ". . . to see description of FINPLAN" NIL)
      (WINCURS '(21 25) NIL)
      (SEL '2 ". . . to begin immediate consultation" NIL)
      (WINCURS '(23 25) NIL)
      AGAIN
        (CLRBFI)(SETQ X (TYI))
        (COND
          ((EQUAL X "1")(DESCRIP))
          ((EQUAL X "2") NIL)
          (T (BEEP)(GO AGAIN)))
        WINCLR)
      (WINCURS '(1 20) NIL)(PRINC "CHOOSE: ")
      (SEL 'L "LIFE-INSURANCE (or both)" NIL)(WINCURS '(3 29) NIL)
      (SEL 'I "INVESTMENT PORTFOLIO" NIL)(WINCURS '(5 29) NIL)
      CHOOSE
        (CLRBFI)(SETQ X (TYI))

```

```

(COND
  ((OR (EQUAL X "L") (EQUAL X "1")) NIL)
  ((OR (EQUAL X "I") (EQUAL X "i")) (SETQ MODE 'INV-MODE))
  (T (BEEP) (GO CHOOSE)))
(WINCLR) (WAIT)
(DEF 'DESCRIP NIL) (DEF 'SEL NIL) (DEF 'CONT NIL)))

(DEF 'SEL
  '[[LAMBDA (NO PHRASE WIND)
    [PROG (POS)
      (SETQ POS (WINCURS NIL WIND)) (WINPOS POS W7)
      (PRINC NO W7)
      (WINCURS (LIST (CAR POS) (ADD1 (CADR POS))) WIND)
      (PRINC " " WIND) (PRINC PHRASE WIND)]]

    (DEF 'CONT
      '[[LAMBDA ()
        (WINATTR 120 W3)
        (PRINTC " Strike any key to continue . . ." W3)
        (CLRBFI) (TYI W3) (WINCLR)])]

    (DEF 'DESCRIP
      '[[LAMBDA ()
        [PROG (POS)
          (WINCLR) (PRINTC " ") (SETQ POS (WINCURS NIL NIL)) (WINPOS POS W7) (WINCLR W7)
          (WINCURS (LIST (CAR POS) (ADD1 (CADR POS)))) NIL)
          (PRINC " FINPLAN is a prototype expert system that provides advice and")
          (PRINC "consultation in developing a personal financial plan suited to your")
          (PRINC "needs. Basic information pertaining to your financial situation")
          (PRINC "temperament, and goals is requested from you through interactive")
          (PRINC "interviews. FINPLAN uses this information to deduce, after a series")
          (PRINC "of inferences, recommendations to be included in your financial plan.")
          (PRINTC " ") (PRINTC " ") (SETQ POS (WINCURS NIL NIL)) (WINPOS POS W7) (WINCLR W7)
          (WINCURS (LIST (CAR POS) (ADD1 (CADR POS)))) NIL)
          (PRINC " An attractive feature incorporated in FINPLAN is the ability to ask")
          (PRINC "for an explanation of recommendations. All lines of reasoning behind")
          (PRINC "a recommendation can be displayed upon request for examination and")
          (PRINC "analysis by you.")
          (PRINTC " ") (PRINTC " ") (SETQ POS (WINCURS NIL NIL)) (WINPOS POS W7) (WINCLR W7)
          (WINCURS (LIST (CAR POS) (ADD1 (CADR POS)))) NIL)
          (PRINC " Because FINPLAN is only a prototype system, it does not address all")
          (PRINTC "relevant tasks of financial planning. Currently, two major tasks are")
          (PRINC "included: recommending an appropriate amount of LIFE-INSURANCE and")
          (PRINC "recommending an INVESTMENT PORTFOLIO. These two tasks are sufficient")
          (PRINC "because the same methodologies developed to handle them could also ")
          (PRINC "be used to handle any other task that might be included in a more")
          (PRINC "comprehensive system. Some additional tasks might include an ")
          (CONT)
          (PRINTC "analysis of and recommendations for net worth, budgeting, and ")
          (PRINC "possibly other forms of insurance. Although portfolio recommendatio")
          ns")
          (PRINC "are reasonably valid in the prototype, the portfolio knowledge bases")
          (PRINTC "also should be expanded and improved in a more comprehensive system")
          (PRINTC " ") (PRINTC " ") (SETQ POS (WINCURS NIL NIL)) (WINPOS POS W7) (WINCLR W7)
          (WINCURS (LIST (CAR POS) (ADD1 (CADR POS)))) NIL)
          (PRINC " FINPLAN is simple and easy to use. You can be a novice or expert in")
          (PRINTC "personal computing or financial planning and still operate the system")
          (PRINTC "easily and effectively. All instructions are self-explanatory. A ")
          (PRINTC "few points, however, should be noted:")(PRINTC " ")
          (PRINTC " - All interview questions require you to press the RETURN ")
          (PRINTC " key after typing your answer. Until you press the RETURN ")
          (PRINTC " key you may change your answer by using the BACKSPACE key.")
          (PRINTC " ")
        ]
      ]
    ]
  ]
)

```

(PRINTC : b - FINPLAN occasionally appears to stop working. It will")
(PRINTC : momentarily suspend execution while performing a garbage")
(PRINTC : collection, which is a normal function inherent to IQLISP.")
(PRINTC : In a few moments the blinking cursor will appear again ")
(PRINTC : and you may proceed.")
(CONT)))

FILE: INVEST.LSP

```
(DEF 'INVEST-RULELIST
  '(LAMBDA ()
    (SETQ INVESTMENT-RULES
      '(((RULE 601
          (IF (OBJECTIVE OF RETIREMENT SECURITY IS SATISFIED . 10)
              (FINANCIAL RISK IS MODERATE . 20)
              (LIQUIDITY IS MODERATE . 4)
              (TAX ADVANTAGE IS MODERATE . 8)
              (INFLATION PROTECTION IS MODERATE . 6)
              (EFFORT REQUIRED IS LOW . 2)
              (SKILL REQUIRED IS NOT DEMANDING . 2))
          (THEN (BLUE CHIP STOCKS . 0))))
        (RULE 602
          (IF (OBJECTIVE OF GROWTH IS SATISFIED . 15)
              (FINANCIAL RISK IS MODERATE . 20)
              (LIQUIDITY IS MODERATE . 4)
              (TAX ADVANTAGE IS MODERATE . 8)
              (INFLATION PROTECTION IS MODERATE . 6)
              (EFFORT REQUIRED IS LOW . 2)
              (SKILL REQUIRED IS NOT DEMANDING . 2))
          (THEN (ESTABLISHED GROWTH STOCKS . 0)))
        (RULE 603
          (IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
              (FINANCIAL RISK IS MODERATE . 20)
              (LIQUIDITY IS MODERATE . 4)
              (TAX ADVANTAGE IS MODERATE . 8)
              (INFLATION PROTECTION IS LOW . 9)
              (EFFORT REQUIRED IS MODERATE . 4)
              (SKILL REQUIRED IS NOT DEMANDING . 2))
          (THEN (ESTABLISHED DIVIDEND STOCKS . 0)))
        (RULE 604
          (IF (OBJECTIVE OF SPECULATION IS SATISFIED . 20)
              (FINANCIAL RISK IS HIGH . 30)
              (LIQUIDITY IS LOW . 6)
              (TAX ADVANTAGE IS MODERATE . 8)
              (INFLATION PROTECTION IS MODERATE . 6)
              (EFFORT REQUIRED IS MODERATE . 4)
              (SKILL REQUIRED IS MODERATE . 4))
          (THEN (NEW ISSUE STOCKS . 0)))
        (RULE 605
          (IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
              (FINANCIAL RISK IS LOW . 10)
              (LIQUIDITY IS MODERATE . 4)
              (TAX ADVANTAGE IS MODERATE . 8)
              (INFLATION PROTECTION IS LOW . 9)
              (EFFORT REQUIRED IS LOW . 2)
              (SKILL REQUIRED IS NOT DEMANDING . 2))
          (THEN (HIGH GRADE CORPORATE BONDS . 0)))
        (RULE 606
          (IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
              (FINANCIAL RISK IS HIGH . 30)
              (LIQUIDITY IS MODERATE . 4)
              (TAX ADVANTAGE IS MODERATE . 6)
              (INFLATION PROTECTION IS LOW . 9)
              (EFFORT REQUIRED IS MODERATE . 4)
              (SKILL REQUIRED IS MODERATE . 4))
          (THEN (LOW GRADE CORPORATE BONDS . 0))))
```

(RULE 607
 (IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
 (FINANCIAL RISK IS LOW . 10)
 (liquidity is moderate . 4)
 (TAX ADVANTAGE IS MODERATE . 8)
 (INFLATION PROTECTION IS LOW . 9)
 (EFFORT REQUIRED IS LOW . 2)
 (SKILL REQUIRED IS NOT DEMANDING . 2))
 (THEN (U S TREASURY BONDS . 0)))

(RULE 608
 (IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
 (FINANCIAL RISK IS LOW . 10)
 (liquidity is moderate . 4)
 (TAX ADVANTAGE IS HIGH . 4)
 (INFLATION PROTECTION IS LOW . 9)
 (EFFORT REQUIRED IS LOW . 2)
 (SKILL REQUIRED IS NOT DEMANDING . 2))
 (THEN (HIGH GRADE MUNICIPALS . 0)))

(RULE 609
 (IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
 (FINANCIAL RISK IS HIGH . 30)
 (liquidity is moderate . 4)
 (TAX ADVANTAGE IS HIGH . 4)
 (INFLATION PROTECTION IS LOW . 9)
 (EFFORT REQUIRED IS MODERATE . 4)
 (SKILL REQUIRED IS MODERATE . 4))
 (THEN (LOW GRADE MUNICIPALS . 0)))

(RULE 610
 (IF (OBJECTIVE OF GROWTH IS SATISFIED . 15)
 (FINANCIAL RISK IS MODERATE . 20)
 (liquidity is moderate . 4)
 (TAX ADVANTAGE IS MODERATE . 8)
 (INFLATION PROTECTION IS MODERATE . 6)
 (EFFORT REQUIRED IS LOW . 2)
 (SKILL REQUIRED IS NOT DEMANDING . 2))
 (THEN (GROWTH-ORIENTED MUTUAL FUNDS . 0)))

(RULE 611
 (IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
 (FINANCIAL RISK IS MODERATE . 20)
 (liquidity is moderate . 4)
 (TAX ADVANTAGE IS MODERATE . 8)
 (INFLATION PROTECTION IS LOW . 9)
 (EFFORT REQUIRED IS LOW . 2)
 (SKILL REQUIRED IS NOT DEMANDING . 2))
 (THEN (INCOME-ORIENTED MUTUAL FUNDS . 0)))

(RULE 612
 (IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
 (FINANCIAL RISK IS LOW . 10)
 (liquidity is high . 2)
 (TAX ADVANTAGE IS LOW . 12)
 (INFLATION PROTECTION IS LOW . 9)
 (EFFORT REQUIRED IS LOW . 2)
 (SKILL REQUIRED IS NOT DEMANDING . 2))
 (THEN (MONEY MARKET ACCOUNT . 0)))

(RULE 613
 (IF (OBJECTIVE OF GROWTH IS SATISFIED . 15)
 (FINANCIAL RISK IS MODERATE . 20)
 (liquidity is low . 6)
 (TAX ADVANTAGE IS LOW . 12)
 (INFLATION PROTECTION IS MODERATE . 6)
 (EFFORT REQUIRED IS HIGH . 6)
 (SKILL REQUIRED IS MODERATE . 4))
 (THEN (SELL COVERED OPTION CALLS . 0)))

(RULE 614
 (IF (OBJECTIVE OF SPECULATION IS SATISFIED . 20)
 (FINANCIAL RISK IS HIGH . 30)
 (LIQUIDITY IS LOW . 6)
 (TAX ADVANTAGE IS LOW . 12)
 (INFLATION PROTECTION IS LOW . 9)
 (EFFORT REQUIRED IS HIGH . 6)
 (SKILL REQUIRED IS MODERATE . 4))
 (THEN (SELL OPEN OPTION CALLS . 0)))

(RULE 615
 (IF (OBJECTIVE OF GROWTH IS SATISFIED . 15)
 (FINANCIAL RISK IS HIGH . 30)
 (LIQUIDITY IS LOW . 6)
 (TAX ADVANTAGE IS MODERATE . 8)
 (INFLATION PROTECTION IS LOW . 9)
 (EFFORT REQUIRED IS HIGH . 6)
 (SKILL REQUIRED IS MODERATE . 4))
 (THEN (BUY OPTIONS/WARRANTS . 0)))

(RULE 616
 (IF (OBJECTIVE OF SPECULATION IS SATISFIED . 20)
 (FINANCIAL RISK IS HIGH . 30)
 (LIQUIDITY IS LOW . 6)
 (TAX ADVANTAGE IS MODERATE . 3)
 (INFLATION PROTECTION IS LOW . 9)
 (EFFORT REQUIRED IS HIGH . 6)
 (SKILL REQUIRED IS HIGH . 6))
 (THEN (FUTURES TRADING . 0)))

(RULE 617
 (IF (OBJECTIVE OF RETIREMENT SECURITY IS SATISFIED . 10)
 (FINANCIAL RISK IS LOW . 10)
 (LIQUIDITY IS LOW . 6)
 (TAX ADVANTAGE IS HIGH . 4)
 (INFLATION PROTECTION IS MODERATE . 6)
 (EFFORT REQUIRED IS LOW . 2)
 (SKILL REQUIRED IS NOT DEMANDING . 2))
 (THEN (INDIVIDUAL RETIREMENT ACCOUNT . 0)))

(RULE 618
 (IF (OBJECTIVE OF GROWTH IS SATISFIED . 15)
 (FINANCIAL RISK IS HIGH . 30)
 (LIQUIDITY IS LOW . 6)
 (TAX ADVANTAGE IS HIGH . 4)
 (INFLATION PROTECTION IS HIGH . 3)
 (EFFORT REQUIRED IS LOW . 2)
 (SKILL REQUIRED IS MODERATE . 4))
 (THEN (TAX "SHELTERS" - REAL ESTATE, BOOKS, ART ETC . 0)))

(RULE 619
 (IF (OBJECTIVE OF GROWTH IS SATISFIED . 15)
 (FINANCIAL RISK IS MODERATE . 20)
 (LIQUIDITY IS LOW . 6)
 (TAX ADVANTAGE IS HIGH . 4)
 (INFLATION PROTECTION IS HIGH . 3)
 (EFFORT REQUIRED IS MODERATE . 4)
 (SKILL REQUIRED IS MODERATE . 4))
 (THEN (REAL ESTATE INVESTMENT TRUST . 0)))

(RULE 620
 (IF (OBJECTIVE OF GROWTH IS SATISFIED . 15)
 (FINANCIAL RISK IS MODERATE . 20)
 (LIQUIDITY IS LOW . 6)
 (TAX ADVANTAGE IS HIGH . 4)
 (INFLATION PROTECTION IS HIGH . 3)
 (EFFORT REQUIRED IS HIGH . 6)
 (SKILL REQUIRED IS MODERATE . 4))
 (THEN (INCOME-PRODUCING REAL ESTATE . 0)))

(RULE 621
(IF (OBJECTIVE OF SPECULATION IS SATISFIED . 20)
(FINANCIAL RISK IS HIGH . 30)
(LIQUIDITY IS LOW . 6)
(TAX ADVANTAGE IS HIGH . 4)
(INFLATION PROTECTION IS HIGH . 3)
(EFFORT REQUIRED IS LOW . 2)
(SKILL REQUIRED IS MODERATE . 4))
(THEN (TAX "SHELTERS" - OIL, GAS, ETC . 0)))

(RULE 622
(IF (OBJECTIVE OF SPECULATION IS SATISFIED . 20)
(FINANCIAL RISK IS HIGH . 30)
(LIQUIDITY IS MODERATE . 4)
(TAX ADVANTAGE IS MODERATE . 8)
(INFLATION PROTECTION IS LOW . 9)
(EFFORT REQUIRED IS LOW . 2)
(SKILL REQUIRED IS NOT DEMANDING . 2))
(THEN (PROFESSIONALLY MANAGED OPTION AND COMMODITY ACCOUNTS . 0)))

(RULE 633
(IF (OBJECTIVE OF RETIREMENT SECURITY IS SATISFIED . 10)
(FINANCIAL RISK IS LOW . 10)
(LIQUIDITY IS LOW . 6)
(TAX ADVANTAGE IS HIGH . 4)
(INFLATION PROTECTION IS LOW . 9)
(EFFORT REQUIRED IS LOW . 2)
(SKILL REQUIRED IS NOT DEMANDING . 2))
(THEN (DEFERRED ANNUITIES OR WHOLE LIFE INSURANCE . 0)))

(RULE 634
(IF (OBJECTIVE OF INCOME IS SATISFIED . 5)
(FINANCIAL RISK IS LOW . 10)
(LIQUIDITY IS MODERATE . 4)
(TAX ADVANTAGE IS MODERATE . 5)
(INFLATION PROTECTION IS LOW . 9)
(EFFORT REQUIRED IS LOW . 2)
(SKILL REQUIRED IS NOT DEMANDING . 2))
(THEN (ANNUITIES . 0)))

(RULE 635
(IF (OBJECTIVE OF GROWTH IS SATISFIED . 15)
(FINANCIAL RISK IS LOW . 10)
(LIQUIDITY IS LOW . 6)
(TAX ADVANTAGE IS HIGH . 4)
(INFLATION PROTECTION IS LOW . 9)
(EFFORT REQUIRED IS LOW . 2)
(SKILL REQUIRED IS NOT DEMANDING . 2))
(THEN (DEFERRED ANNUITIES . 0))))])

FILE: I-PAR.LSP

```

(DEF 'INV-PARAM
  '(LAMBDA ()
    (PROG (DIVERS-EXCEPT DIVERS-DEFAULT OTHER-PARAM)
      (SETQ DIVERS-EXCEPT
        '((RULE 1 (IF (YOU ARE A CONSERVATIVE RISK-TAKER))
          (THEN (YOUR PORTFOLIO SHOULD BE CONSERVATIVE)))
        (RULE 2 (IF (YOU ARE A MODERATE RISK-TAKER))
          (THEN (YOUR PORTFOLIO SHOULD BE MODERATELY RISKY)))
        (RULE 3 (IF (YOU ARE AN AGGRESSIVE RISK-TAKER))
          (THEN (YOUR PORTFOLIO SHOULD BE AGGRESSIVE)))
        (RULE 8 (IF (YOUR PORTFOLIO SHOULD BE CONSERVATIVE)
          (DESIRE TO PARTICIPATE IS LOW)
          (AVAILABLE CAPITAL IS $6000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE LOW)))
        (RULE 9 (IF (YOUR PORTFOLIO SHOULD BE MODERATELY RISKY)
          (DESIRE TO PARTICIPATE IS LOW)
          (AVAILABLE CAPITAL IS $6000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE LOW)))
        (RULE 10 (IF (YOUR PORTFOLIO SHOULD BE AGGRESSIVE)
          (DESIRE TO PARTICIPATE IS LOW)
          (AVAILABLE CAPITAL IS $9000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE MODERATE)))
        (RULE 11 (IF (YOUR PORTFOLIO SHOULD BE CONSERVATIVE)
          (DESIRE TO PARTICIPATE IS MODERATE)
          (AVAILABLE CAPITAL IS $6000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE LOW)))
        (RULE 12 (IF (YOUR PORTFOLIO SHOULD BE MODERATELY RISKY)
          (DESIRE TO PARTICIPATE IS MODERATE)
          (AVAILABLE CAPITAL IS $9000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE MODERATE)))
        (RULE 13 (IF (YOUR PORTFOLIO SHOULD BE AGGRESSIVE)
          (DESIRE TO PARTICIPATE IS MODERATE)
          (AVAILABLE CAPITAL IS $12000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE HIGH)))
        (RULE 14 (IF (YOUR PORTFOLIO SHOULD BE CONSERVATIVE)
          (DESIRE TO PARTICIPATE IS HIGH)
          (AVAILABLE CAPITAL IS $9000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE MODERATE)))
        (RULE 15 (IF (YOUR PORTFOLIO SHOULD BE MODERATELY RISKY)
          (DESIRE TO PARTICIPATE IS HIGH)
          (AVAILABLE CAPITAL IS $12000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE HIGH)))
        (RULE 16 (IF (YOUR PORTFOLIO SHOULD BE AGGRESSIVE)
          (DESIRE TO PARTICIPATE IS HIGH)
          (AVAILABLE CAPITAL IS $12000 OR MORE))
          (THEN (DIVERSIFICATION CAN BE HIGH)))))

      (SETQ DIVERS-DEFAULT
        '((RULE 1 (IF (YOU ARE A CONSERVATIVE RISK-TAKER))
          (THEN (YOUR PORTFOLIO SHOULD BE CONSERVATIVE)))
        (RULE 2 (IF (YOU ARE A MODERATE RISK-TAKER))
          (THEN (YOUR PORTFOLIO SHOULD BE MODERATELY RISKY)))
        (RULE 3 (IF (YOU ARE AN AGGRESSIVE RISK-TAKER))
          (THEN (YOUR PORTFOLIO SHOULD BE AGGRESSIVE)))
        (RULE 4 (IF (TRY TO INVEST AT LEAST $3000 IN EACH CLASS)
          (AVAILABLE CAPITAL IS LESS THAN $6000))
          (THEN (DIVERSIFICATION CAN BE MINIMUM))))
```

AD-A147 716 FINPLAN: A PROTOTYPE EXPERT SYSTEM FOR PERSONAL
FINANCIAL PLANNING(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH J D PINC AUG 84

2/2

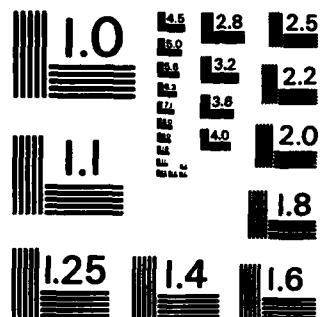
UNCLASSIFIED AFIT/CI/NR-84-67T

F/G 9/1

NL

END

RECORDED
SERIALIZED
FILED
OFC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

```

(RULE 5 (IF (TRY TO INVEST AT LEAST $3000 IN EACH CLASS)
           (AVAILABLE CAPITAL IS BETWEEN $6000 AND $9000))
           (THEN (DIVERSIFICATION CAN BE LOW)))
(RULE 6 (IF (TRY TO INVEST AT LEAST $3000 IN EACH CLASS)
           (AVAILABLE CAPITAL IS BETWEEN $9000 AND $12000))
           (THEN (DIVERSIFICATION CAN BE MODERATE)))
(RULE 7 (IF (TRY TO INVEST AT LEAST $3000 IN EACH CLASS)
           (AVAILABLE CAPITAL IS $12000 OR MORE))
           (THEN (DIVERSIFICATION CAN BE HIGH)))))

(SETQ OTHER-PARAM
'((RULE 17 (IF (DIVERSIFICATION CAN BE MINIMUM))
           (THEN (ONE INVESTMENT CLASS IS ENOUGH)))
(RULE 18 (IF (DIVERSIFICATION CAN BE LOW))
           (THEN (TWO INVESTMENT CLASSES ARE ADEQUATE)))
(RULE 19 (IF (DIVERSIFICATION CAN BE MODERATE))
           (THEN (THREE INVESTMENT CLASSES ARE ADEQUATE)))
(RULE 20 (IF (DIVERSIFICATION CAN BE HIGH))
           (THEN (FOUR INVESTMENT CLASSES ARE ADEQUATE)))
(RULE 21 (IF (ONE INVESTMENT CLASS IS ENOUGH))
           (THEN (INVEST ALL OF ORIGINAL CAPITAL IN ONE CLASS)))
(RULE 22 (IF (TWO INVESTMENT CLASSES ARE ADEQUATE))
           (THEN (INVEST ABOUT 50% OF ORIGINAL CAPITAL IN EACH CLASS)))
(RULE 23 (IF (THREE INVESTMENT CLASSES ARE ADEQUATE))
           (THEN (INVEST ABOUT 33% OF ORIGINAL CAPITAL IN EACH CLASS)))
(RULE 24 (IF (FOUR INVESTMENT CLASSES ARE ADEQUATE))
           (THEN (INVEST ABOUT 25% OF ORIGINAL CAPITAL IN EACH CLASS)))))

(ADD '(TRY TO INVEST AT LEAST $3000 IN EACH CLASS))
(COND ((LT CAPITAL 6000)
           (ADD '(AVAILABLE CAPITAL IS LESS THAN $6000))))
(COND ((GE CAPITAL 6000)
           (ADD '(AVAILABLE CAPITAL IS $6000 OR MORE))))
(COND ((GE CAPITAL 9000)
           (ADD '(AVAILABLE CAPITAL IS $9000 OR MORE))))
(COND ((GE CAPITAL 12000)
           (ADD '(AVAILABLE CAPITAL IS $12000 OR MORE))))
(COND ((AND (GE CAPITAL 6000)(LT CAPITAL 9000))
           (ADD '(AVAILABLE CAPITAL IS BETWEEN $6000 AND $9000)))
(COND ((AND (GE CAPITAL 9000)(LT CAPITAL 12000))
           (ADD '(AVAILABLE CAPITAL IS BETWEEN $9000 AND $12000)))
(FORWARDCHAIN DIVERS-EXCEPT)
  (SETQ DIVERS-EXCEPT NIL)
(COND
  ((NOT (FIND '(DIVERSIFICATION +)))(FORWARDCHAIN DIVERS-DEFAULT)))
(FORWARDCHAIN OTHER-PARAM)
  (SETQ DIVERS-DEFAULT NIL)(SETQ OTHER-PARAM NIL)
(COND
  ((RECALL '(ONE INVESTMENT CLASS IS ENOUGH))
    (SETQ UNIT CAPITAL)(SETQ NO-INV 1))
  ((RECALL '(TWO INVESTMENT CLASSES ARE ADEQUATE))
    (SETQ UNIT (FIX (/ CAPITAL 2)))(SETQ NO-INV 2))
  ((RECALL '(THREE INVESTMENT CLASSES ARE ADEQUATE))
    (SETQ UNIT (FIX (/ CAPITAL 3)))(SETQ NO-INV 3))
  ((RECALL '(FOUR INVESTMENT CLASSES ARE ADEQUATE))
    (SETQ UNIT (FIX (/ CAPITAL 4)))(SETQ NO-INV 4)))
  (SETQ UNIT (+ (/ UNIT 1000) 1000))
  (SETQ REMAIN (% CAPITAL UNIT))))
```

FILE: I-RULE.LSP

```

(DEF 'INS-RULELIST
  '(LAMBDA ()
    (PROG (INS-RULES)
      (ADD '(YOU WANT TO ESTIMATE LIFE INSURANCE NEEDS))
      (ADD '(THIS ESTIMATE WAS MADE BY BAILARD, BIEHL & KAISER INC))
      (ADD '(THIS AVERAGE REFLECTS BENEFITS FOR MINORS AND FUTURE RETIREMENT))
      (ADD '(THIS COST IS BASED ON AMERICAN COUNCIL ON EDUCATIONS ESTIMATE))
      (ADD '(ASSUME A 4 YEAR EDUCATION FOR EACH CHILD))
      (SETQ INS-RULES
        '(
          (RULE 1
            (IF (YOU WANT TO ESTIMATE LIFE INSURANCE NEEDS))
            (THEN (YOU MUST ADD IT IN ORDER TO ESTIMATE TOTAL MONETARY NEEDS)
                  (YOU MUST CONSIDER TOTAL MONETARY NEEDS)
                  (YOU MUST CONSIDER IT IN ORDER TO ESTIMATE TOTAL MONETARY NEEDS)
                  (LIFE INSURANCE NEEDS SHOULD BE ESTIMATED)
                  (SUBTRACT INVESTMENT ASSETS IN HAND FROM TOTAL MONETARY NEEDS)))
          (RULE 2
            (IF (YOU MUST ADD IT IN ORDER TO ESTIMATE TOTAL MONETARY NEEDS))
            (THEN (THIS COST SHOULD BE PROVIDED FOR AT TIME OF DEATH)))
          (RULE 3
            (IF (THIS COST SHOULD BE PROVIDED FOR AT TIME OF DEATH))
            (THEN (YOU MUST INCLUDE FUNERAL EXPENSES IN TOTAL MONETARY NEEDS)
                  (YOU MUST INCLUDE NON-MORTGAGE DEBT IN TOTAL MONETARY NEEDS)
                  (YOU MUST INCLUDE EMERGENCY FUND REQUIREMENT IN TOTAL MONETARY NEEDS)
                  (YOU MUST INCLUDE COLLEGE FUND REQUIREMENT IN TOTAL MONETARY NEEDS)
                  (YOU MUST INCLUDE TOTAL LIVING EXPENSES IN TOTAL MONETARY NEEDS)))
          (RULE 4
            (IF (THIS COST IS BASED ON AMERICAN COUNCIL ON EDUCATIONS ESTIMATE)
                (YOU HAVE A PRIVATE COLLEGE IN MIND))
            (THEN (FIGURE $9000 A YEAR PER CHILD)))
          (RULE 5
            (IF (THIS COST IS BASED ON AMERICAN COUNCIL ON EDUCATIONS ESTIMATE)
                (YOU HAVE A PUBLIC COLLEGE IN MIND))
            (THEN (FIGURE $5000 A YEAR PER CHILD)))
          (RULE 6
            (IF (YOU DO NOT INTEND OR NEED TO PROVIDE A COLLEGE FUND))
            (THEN (THERE IS NO NEED FOR PROVIDING A COLLEGE FUND)))
          (RULE 7
            (IF (YOU INTEND TO PROVIDE A COLLEGE FUND)
                (FIGURE $5000 A YEAR PER CHILD)
                (ASSUME A 4 YEAR EDUCATION FOR EACH CHILD)
                (YOU MUST INCLUDE COLLEGE FUND REQUIREMENT IN TOTAL MONETARY NEEDS)))
          (S))
            (THEN (THERE IS A NEED FOR PROVIDING A COLLEGE FUND)))
          (RULE 8
            (IF (YOU INTEND TO PROVIDE A COLLEGE FUND)
                (FIGURE $9000 A YEAR PER CHILD)
                (ASSUME A 4 YEAR EDUCATION FOR EACH CHILD)
                (YOU MUST INCLUDE COLLEGE FUND REQUIREMENT IN TOTAL MONETARY NEEDS)))
          (S))
            (THEN (THERE IS A NEED FOR PROVIDING A COLLEGE FUND)))
          (RULE ?
            (IF (YOU MUST INCLUDE FUNERAL EXPENSES IN TOTAL MONETARY NEEDS)
                (THIS ESTIMATE WAS MADE BY BAILARD, BIEHL & KAISER INC)
                (YOUR GROSS ESTATE IS UNDER $20000)))
    )
  )
)
```

(THEN (YOUR FUNERAL EXPENSES ARE ESTIMATED TO BE \$2200)))
(RULE 10
 (IF (YOU MUST INCLUDE FUNERAL EXPENSES IN TOTAL MONETARY NEEDS)
 (THIS ESTIMATE WAS MADE BY BAILARD, BIEHL & KAISER INC)
 (YOUR GROSS ESTATE IS BETWEEN \$20000 AND \$200000))
 (THEN (YOUR FUNERAL EXPENSES ARE ESTIMATED TO BE \$5000)))
(RULE 11
 (IF (YOU MUST INCLUDE FUNERAL EXPENSES IN TOTAL MONETARY NEEDS)
 (THIS ESTIMATE WAS MADE BY BAILARD, BIEHL & KAISER INC)
 (YOUR GROSS ESTATE IS MORE THAN \$200000))
 (THEN (YOUR FUNERAL EXPENSES ARE ESTIMATED TO BE \$10000)))
(RULE 12
 (IF (YOU MUST INCLUDE TOTAL LIVING EXPENSES IN TOTAL MONETARY NEEDS))
 (THEN (YOU MUST CONSIDER IT IN ORDER TO ESTIMATE TOTAL LIVING EXPENSES)))
(RULE 13
 (IF (YOU MUST CONSIDER IT IN ORDER TO ESTIMATE TOTAL LIVING EXPENSES))
 (THEN (AVERAGE ANNUAL LIVING EXPENSES ARE CONSIDERED)
 (SPOUSES INCOME IS SUBTRACTED FROM ANNUAL LIVING EXPENSES)
 (SOCIAL SECURITY BENEFITS ARE SUBTRACTED FROM ANNUAL LIVING EXPENSES))
 (NET ANNUAL LIVING EXPENSES ARE CONSIDERED)))
(RULE 14
 (IF (THIS AVERAGE REFLECTS BENEFITS FOR MINORS AND FUTURE RETIREMENT)
 (YOU HAVE TWO OR MORE MINOR CHILDREN)
 (THIS ESTIMATE WAS MADE BY BAILARD, BIEHL & KAISER INC)
 (SOCIAL SECURITY BENEFITS ARE SUBTRACTED FROM ANNUAL LIVING EXPENSES))
 (THEN (SOCIAL SECURITY BENEFITS WILL BE \$5000 PER YEAR)))
(RULE 141
 (IF (THIS AVERAGE REFLECTS BENEFITS FOR MINORS AND FUTURE RETIREMENT)
 (YOU HAVE ONE MINOR CHILD)
 (THIS ESTIMATE WAS MADE BY BAILARD, BIEHL & KAISER INC)
 (SOCIAL SECURITY BENEFITS ARE SUBTRACTED FROM ANNUAL LIVING EXPENSES))
 (THEN (SOCIAL SECURITY BENEFITS WILL BE \$4000 PER YEAR)))
(RULE 142
 (IF (THIS AVERAGE REFLECTS BENEFITS FOR MINORS AND FUTURE RETIREMENT)
 (YOU HAVE NO MINOR CHILDREN)
 (THIS ESTIMATE WAS MADE BY BAILARD, BIEHL & KAISER INC)
 (SOCIAL SECURITY BENEFITS ARE SUBTRACTED FROM ANNUAL LIVING EXPENSES))
 (THEN (SOCIAL SECURITY BENEFITS WILL BE \$3000 PER YEAR)))
(RULE 15
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 25 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN A CONSERVATIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 20)))
(RULE 16
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 30 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN A CONSERVATIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 22)))
(RULE 17
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 35 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN A CONSERVATIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 25)))
(RULE 18
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 40 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN A CONSERVATIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 27)))
(RULE 19
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 45 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN A CONSERVATIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 30)))

(RULE 20
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 30 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN A CONSERVATIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 31)))
(RULE 21
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 35 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN A CONSERVATIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 33)))
(RULE 22
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 40 YEARS OR MORE)
 (INSURANCE PROCEEDS WILL BE INVESTED IN A CONSERVATIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 35)))
(RULE 23
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 25 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN AN AGGRESSIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 16)))
(RULE 24
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 30 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN AN AGGRESSIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 17)))
(RULE 25
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 35 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN AN AGGRESSIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 19)))
(RULE 26
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 40 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN AN AGGRESSIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 20)))
(RULE 27
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 45 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN AN AGGRESSIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 21)))
(RULE 28
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 50 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN AN AGGRESSIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 21)))
(RULE 29
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 55 YEARS)
 (INSURANCE PROCEEDS WILL BE INVESTED IN AN AGGRESSIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 22)))
(RULE 30
 (IF (YOUR SPOUSE WILL BE 90 IN ABOUT 60 YEARS OR MORE)
 (INSURANCE PROCEEDS WILL BE INVESTED IN AN AGGRESSIVE PORTFOLIO))
 (THEN (INVESTMENT RATE FACTOR IS 23))))

(FORWARDCHAIN INS-RULES))))

FILE: LOAD.LSP

```
<DEF 'LOAD
  '(LAMBDA (FILE)
    (PROG (X Y)
      (SETQ X
        (INPUT FILE))
    LOOP
      (SETQ Y (READ X))
      (COND
        [(NOT (EQUAL Y (EOF))) (EVAL Y)
         (GO LOOP)])))
```

FILE: QUERY.LSP

```
(DEF 'INV-INTERVIEW
  '(LAMBDA ()
    (PROG ()
      (WINCLR W1)
      (PRINTC "In order to offer recommendations regarding an investment"
portfolio" W1)
      (PRINTC "for you, please answer the following questions." W1)
      (PRINTC " " W1)
      AGE
      (PRINTC "What is your age (in years)?" W1)
      (PRINTC " " W1)(PRINTC " " W1)
      (SETQ X
            (READ W1))
      (COND
        [(NOT (NUMBERP X))
         (BEEP)
         (GO AGE)]
        [(

```

```

((EQUAL X 2)
  (ADD '(YOU INDICATED AN OBJECTIVE OF INCOME)))
((EQUAL X 3)
  (ADD '(YOU INDICATED AN OBJECTIVE OF GROWTH)))
((EQUAL X 4)
  (ADD '(YOU INDICATED AN OBJECTIVE OF SPECULATION)))
[IT (BEEP)(GO OBJ)])
(WINCLR W1)
EXP
  (PRINTC "What is your experience in trading securities?" W1)
(ANS '1 "Never traded" W1)
(ANS '2 "Less than one year" W1)
(ANS '3 "More than one year" W1)(PRINTC " " W1)(PRINTC " " W1)
(SETQ X
  (READ W1))
(COND
  ((NOT (NUMBERP X))
    (BEEP)
    (GO EXP)))
  ((EQUAL X 1)
    (ADD '(EXPERIENCE LEVEL IS MINIMAL)))
  ((EQUAL X 2)
    (ADD '(EXPERIENCE LEVEL IS MODERATE)))
  ((EQUAL X 3)
    (ADD '(EXPERIENCE LEVEL IS HIGH)))
[IT
  (BEEP)
  (GO EXP)])
(WINCLR W1)
WSJ
  (PRINTC
  "Do you subscribe to the Wall Street Journal or other financial periodicals?" W1)
  (PRINTC "periodicals or services?" W1)
(PRINTC " " W1)(PRINTC " " W1)
(SETQ X
  (READ W1))
(COND
  ((OR (EQUAL (CHRVAL X) 89)(EQUAL (CHRVAL X) 121))
    (ADD '(FINANCIAL SERVICE IS CURRENTLY USED)))
  ((OR (EQUAL (CHRVAL X) 78)(EQUAL (CHRVAL X) 110))
    (ADD '(FINANCIAL SERVICE IS NOT CURRENTLY USED)))
[IT
  (BEEP)
  (GO WSJ)])
(WINCLR W1)
AVAIL
  (PRINTC
  "What is the amount of liquid assets you have available for investment?" W1)
  (PRINTC " " W1)(PRINTC " " W1)
(SETQ CAPITAL
  (READ W1))
(COND
  ((NOT (NUMBERP CAPITAL))
    (BEEP)
    (GO AVAIL)))
  ((GT CAPITAL 1000000)(BEEP)
    (PRINTC "Go hire an accountant." W1)(GO AVAIL))
  [IT (ADD (LIST 'AVAILABLE 'CAPITAL 'IS CAPITAL))])
(WINCLR W1)
SECURITY
  (PRINTC
  "How secure is your income flow on which you depend for living?" W1)

```

```

(PRINTC "expenses?" W1)
(ANS '1 "Not very secure" W1)
(ANS '2 "Moderately secure" W1)
(ANS '3 "Very secure" W1)(PRINTC " " W1)(PRINTC " " W1)
(SETQ X
  (READ W1))
(COND
  [((NOT (NUMBERP X))
    (BEEP)
    (GO SECURE))
   [((EQUAL X 1)
     (ADD '(INCOME IS NOT VERY SECURE)))
    ((EQUAL X 2)
     (ADD '(INCOME IS MODERATELY SECURE)))
    ((EQUAL X 3)
     (ADD '(INCOME IS VERY SECURE)))
   ]
  ]
  [T
    (BEEP)
    (GO SECURE)])
 (WINCLR W1)
EMERGENCY
(PRINTC "What level is your emergency reserve fund?" W1)
(ANS '1 "Inadequate" W1)
(ANS '2 "Marginally adequate" W1)
(ANS '3 "Probably adequate" W1)(PRINTC " " W1)(PRINTC " " W1)
(SETQ X
  (READ W1))
(COND
  [((NOT (NUMBERP X))
    (BEEP)
    (GO EMERGENCY))
   [((EQUAL X 1)
     (ADD '(EMERGENCY RESERVES ARE INADEQUATE)))
    ((EQUAL X 2)
     (ADD '(EMERGENCY RESERVES ARE MODERATELY ADEQUATE)))
    ((EQUAL X 3)
     (ADD '(EMERGENCY RESERVES ARE ADEQUATE)))
   ]
  ]
  [T
    (BEEP)
    (GO EMERGENCY)])
 (WINCLR W1)
TAXABLE
(PRINTC
  "Your estimated taxable income for this year probably will"
  "be . . ." W1)
(ANS '1 "Less than $25,000" W1)
(ANS '2 "Between $25,000 and $45,000" W1)
(ANS '3 "More than $45,000" W1)(PRINTC " " W1)(PRINTC " " W1)
(SETQ X
  (READ W1))
(COND
  [((NOT (NUMBERP X))
    (BEEP)
    (GO TAXABLE))
   [((EQUAL X 1)
     (ADD '(TAXABLE INCOME IS LESS THAN $25000)))
    ((EQUAL X 2)
     (ADD '(TAXABLE INCOME IS BETWEEN $25000 AND $45000)))
    ((EQUAL X 3)
     (ADD '(TAXABLE INCOME IS MORE THAN $45000)))
   ]
  ]
  [T
    (BEEP)
    (GO TAXABLE)])
 (WINCLR W1)

```

```

    (T
      (BEEP)
      (GO TAXABLE))
    (WINCLR W1)

RISK
  (PRINTC
    "How would you rate yourself generally as a risk taker?" W1)
  (ANS '1 " You want to protect your money from loss above all other~
   * W1)
  (PRINTC "           considerations." W1)
  (ANS '2 " You are willing to take some moderate risks to increase~
   your" W1)
  (PRINTC "           investment funds to a greater degree." W1)
  (ANS '3 " You'll accept a high degree of risk for a good chance at~
   a big" W1)
  (PRINTC "           reward." W1)(PRINTC " * W1)(PRINTC "     * W1)
  (SETQ X
    (READ W1))
  (COND
    [((NOT (NUMBERP X))
      (BEEP)
      (GO RISK))
     [((EQUAL X 1)
       (ADD '(YOU ARE A CONSERVATIVE RISK-TAKER)))
      [((EQUAL X 2)
        (ADD '(YOU ARE A MODERATE RISK-TAKER)))
       [((EQUAL X 3)
         (ADD '(YOU ARE AN AGGRESSIVE RISK-TAKER))))]
     [T
      (BEEP)
      (GO RISK))]
    (WINCLR W1))

EFFORT
  (PRINTC
    "To what degree do you wish to actively participate in the" ~
    W1)
  (PRINTC "management of your investments?" W1)
  (ANS '1 " Minimum participation" W1)
  (ANS '2 " Moderate participation" W1)
  (ANS '3 " High degree of participation" W1)
  (PRINTC " * W1)(PRINTC "     * W1)
  (SETQ X
    (READ W1))
  (COND
    [((NOT (NUMBERP X))
      (BEEP)
      (GO EFFORT))
     [((EQUAL X 1)
       (ADD '(DESIRE TO PARTICIPATE IS LOW)))
      [((EQUAL X 2)
        (ADD '(DESIRE TO PARTICIPATE IS MODERATE)))
       [((EQUAL X 3)
         (ADD '(DESIRE TO PARTICIPATE IS HIGH))))]
     [T
      (BEEP)
      (GO EFFORT))]
    (WINCLR W1))

INFLATION
  (PRINTC
    "What are your expectations about inflation over the next 3~
     years?" W1)

```

```
(ANS '1 "Anticipate inflation to be low" W1)
(ANS '2 "Anticipate inflation to be moderate" W1)
(ANS '3 "Anticipate inflation to be high" W1)
(PRINTC " W1)(PRINTC "      " W1)
(SETQ X
  (READ W1))
(COND
  [(NOT (NUMBERP X))
   (BEEP)
   (GO INFLATION)]
  [(EQUAL X 1)
   (ADD '(ANTICIPATED INFLATION IS LOW))]
  [(EQUAL X 2)
   (ADD '(ANTICIPATED INFLATION IS MODERATE))]
  [(EQUAL X 3)
   (ADD '(ANTICIPATED INFLATION IS HIGH))]
  [T
   (BEEP)
   (GO INFLATION)]])
```

FILE: RUN.LSP

```

(DEF 'RUN
  '(LAMBDA ()
    (PROG ()
      (LOAD "B:INIT.LSP")
      (INITIALIZE) (DEF 'INITIALIZE NIL)
      (LOAD "B:INTRO.LSP")
      (INTRO)(DEF 'INTRO NIL)
      (LOAD "B:CONTROL.LSP")
      (COND ((EQUAL MODE 'INV-MODE)(GO INV)))
      (LOAD "B:DO-INS.LSP")
      (DO-INSURANCE) (DEF 'DO-INSURANCE NIL)
      INV
      (LOAD "B:DO-INV.LSP")
      (DO-INVESTMENTS)(WINCLR)]]))

(DEF 'WAIT
  '(LAMBDA ()
    (WINCLR W2)
    (WINATTR 120 W2)
    (PRINC " PLEASE WAIT      " W2)[])

(DEF 'STOP-WAIT
  '(LAMBDA () (WINATTR 7 W2)
  (WINCLR W2)[])

(DEF 'CHOICE
  '(LAMBDA ()
    (PROG ()
      AGAIN
      (SETQ STATUS 'KEEP-GOING)
      (WINCLR W3)
      (WINATTR 120 W3)
      (PRINTC " SELECT: <I>Inquire <C>Continue" W3)
      (CLRBFI)(SETQ X (TYI W3))
      (WINATTR 7 W3)(WINCLR W3)(SETQ WIN W4)
      (COND
        ((OR (EQUAL X "I")(EQUAL X "I"))
         (PROG (NO-REA)
            ANOTHER
            (COND
              ((EQUAL MODE 'INS-MODE)(WINATTR 120 W1)(WINCLR W1)
               (WINCLR W4)(SHOW-INS W4)(INQUIRE-INS))
              ((EQUAL MODE 'CONSTR-MODE)(WINATTR 120 W1)
               (WINCLR W1)(WINCLR W4)(SHOW-CONSTR)(INQUIRE-CONSTR))
              ((EQUAL MODE 'INV-MODE)(WINATTR 120 W1)
               (WINCLR W1)(WINCLR W4)(SHOW-INV)(INQUIRE-INV)))
            (COND
              ((NOT (EQUAL STATUS 'EXIT))(INQUIRE)))
            (COND
              ((EQUAL STATUS 'LDR-DONE)(GO ANOTHER))
              ((EQUAL STATUS 'EXIT)(WINATTR 7 W1)(WINCLR W1)
               (SETQ WIN W1)
               (COND ((EQUAL MODE 'INS-MODE)(SHOW-INS W1))
                     ((EQUAL MODE 'CONSTR-MODE)(SHOW-CONSTR))
                     ((EQUAL MODE 'INV-MODE)(SHOW-INV))))))
          (GO AGAIN))
        ((OR (EQUAL X "C")(EQUAL X "c"))
         NIL)
      IT
      (BEEP)
      (GO AGAIN)]])))

```

FILE: RANK.LSP

```

(DEF 'RANK-INV
  '(LAMBDA ()
    [PROG (INVS INVEST-RULE I RANK)
      (SETQ RANKED-INVS NIL) (SETQ INVS (INVESTMENT-RULELIST))
      (DEF 'INVESTMENT-RULELIST NIL)
      LOOP
        (COND
          [(NULL INVS)(RETURN NIL)]
          [IT (SETQ INVEST-RULE (CAR INVS))
            (SETQ I (CADAR (CDDR INVEST-RULE)))
            (SETQ RANK (EVAL-INV INVEST-RULE))
            (RPLACD (LAST I) RANK))]
          (SETQ INVS (CDR INVS))
          (GO LOOP)]))

(DEF 'EVAL-INV
  '(LAMBDA (RULE)
    [PROG (IFS VAL CHAR NO RK)
      (SETQ IFS (CDR (CADDR RULE))) (SETQ NO 0)
      (SETQ CHAR (ARRAY 1 7)) (SETQ RK 0)
      LOOPIF
        (COND
          [(EQUAL NO 7)(RETURN RK)]
          [IT (STORE (CHAR NO) (CDR (LAST (CAR IFS))))
            (SETQ VAL (- (CHAR NO) (GOAL NO)))
            (COND
              ((PLUSP VAL)(SETQ VAL (* VAL 5)))
              (SETQ VAL (ABS VAL)) (SETQ RK (+ RK VAL))
              (SETQ NO (ADD1 NO))(SETQ IFS (CDR IFS))(GO LOOP[F]))])
        )]

(DEF 'SORT
  '(LAMBDA ()
    [PROG (MIN PTR INUMT)
      (SETQ MIN 500)
      (MAPC 'LAMBDA (X)
        (PROG (I)
          (SETQ I (CADAR (CDDR X)))
          (COND
            ((LT (CDR (LAST I)) MIN)
              (SETQ MIN (CDR (LAST I)))
              (SETQ PTR X))))]
        ) INVESTMENT-RULES)
      (SETQ INUMT (CADAR (CDDR PTR)))
      (RPLACD (LAST INUMT) 500)
      (SETQ RULES-USED (CONS PTR RULES-USED))
      (ADD INUMT) (RETURN INUMT)]))

(DEF 'SORT-INVS
  '(LAMBDA ()
    (SETQ FIRST (SORT))(SETQ SECOND (SORT))
    (SETQ THIRD (SORT))(SETQ FOURTH (SORT)))])

```

BIOGRAPHICAL SKETCH

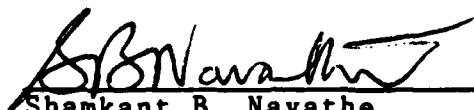
James D. Pinc is a captain on active duty service in the United States Air Force and is attending the University of Florida under the Air Force Institute of Technology program. He received a Bachelor of Science degree majoring in management and finance from the United States Air Force Academy in 1976. He also earned a Master of Business Administration degree in 1980 while attending night school at Western New England College, Springfield, Massachusetts. Before attending the University of Florida, Captain Pinc worked as a computer acquisition cost analyst for five years at Hanscom Air Force Base, Massachusetts. While at Hanscom, he received several citations and awards, including being named the Air Force Communication Command Data Automation Officer of the Year in 1981. For recreation, Captain Pinc enjoys playing golf and ice hockey, and flying as a private pilot. He is married to the former Miss Margo Ceplinskas of Manchester, New Hampshire.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



Douglas D. Dankel, II, Chairman
Assistant Professor of
Computer and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



Shamkant B. Navathe
Associate Professor of
Computer and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



David J. Nye
Associate Professor of
Finance, Insurance, and Real Estate

This thesis was submitted to the Graduate Faculty of
the Department of Computer and Information Sciences in the
College of Business Administration and to the Graduate
School, and was accepted as partial fulfillment of the
requirements for the degree of Master of Science.

August 1984

Dean for Graduate Studies
and Research

END

FILMED

12-84

DTIC